SAM R30

IEEE 802.15.4 Sub-GHz System-in-Package Datasheet

Introduction

The SAM R30 is a series of Ultra low-power microcontrollers equipped with an IEEE® 802.15.4-2003/2006/2011 compliant RF interface for the sub-1GHz frequency bands such as 780 MHz (China), 868 MHz (Europe) and 915 MHz (North America). It uses the 32-bit ARM® Cortex®-M0+ processor at max. 48MHz (2.46 CoreMark®/MHz) and offers 256KB of Flash and 40KB of SRAM in both 32- and 48-pin packages. Sophisticated power management technologies, such as power domain gating, SleepWalking, Ultra low-power peripherals and more, allow for very low current consumptions.

The highly configurable peripherals include a touch controller supporting capacitive interfaces with proximity sensing. The sub-GHz RF interface supports BPSK and O-QPSK modulation schemes according to the IEEE standard and offers output power values of more than +8dBm and receiver sensitivities below -108 dBm.

Features

- Processor
  - ARM Cortex-M0+ CPU running at up to 48 MHz
    - Single-cycle hardware multiplier
    - Micro Trace Buffer (MTB)
- Memories
  - 256 KB in-system self-programmable Flash
  - 32 KB SRAM main memory
  - 8 KB SRAM low power memory
- System
  - Power-on reset (POR) and brown-out detection (BOD)
  - Internal and external clock options with 48 MHz Digital Frequency Locked Loop (DFLL48M) and 48 MHz to 96 MHz Fractional Digital Phase Locked Loop (FDPLL96M)
  - External Interrupt Controller (EIC)
  - Up to 16 external interrupts
  - One non-maskable interrupt
  - Two-pin Serial Wire Debug (SWD) programming, test and debugging interface
- Low Power
  - Idle and standby sleep modes
  - SleepWalking peripherals
- Integrated Ultra Low Power Transceiver for 700/800/900 MHz ISM Band:
  - Chinese WPAN band from 779 to 787 MHz
  - European SRD band from 863 to 870 MHz
  - North American ISM band from 902 to 928 MHz
  - Japanese band from 915 to 930 MHz
- Direct Sequence Spread Spectrum with different modulation and data rates:
  - BPSK with 20 and 40 kb/s, compliant to IEEE® 802.15.4-2003/2006/2011
  - O-QPSK with 100 and 250 kb/s, compliant to IEEE 802.15.4-2006/2011
  - O-QPSK with 250 kb/s, compliant to IEEE 802.15.4-2011

  - O-QPSK with 200, 400, 500 and 1000 kb/s PSDU data rate
  - Industry leading link budget:
    - RX Sensitivity up to -110 dBm
    - TX Output Power up to +11 dBm
  - Hardware Assisted MAC
    - Auto-Acknowledge
    - Auto-Retry
    - CSMA-CA and Listen Before Talk (LBT)
    - Automatic address filtering and automated FCS check
  - Special IEEE 802.15.4™-2011 hardware support:
    - FCS computation and Clear Channel Assessment
    - RSSI measurement, Energy Detection and Link Quality Indication
  - Antenna Diversity and PA/LNA Control
  - 128-Byte TX/RX Frame Buffer
  - Integrated 16 MHz Crystal Oscillator (external crystal needed)
  - Fully integrated, fast settling Transceiver PLL to support Frequency Hopping
  - Hardware Security (AES, True Random Generator)
- Peripherals
  - 16-channel Direct Memory Access Controller (DMAC)
  - 12-channel Event System
  - Up to three 16-bit Timer/Counters (TC), configurable as either:
    - 16-bit TC with compare/capture channels
    - 8-bit TC with compare/capture channels
    - One 32-bit TC with compare/capture channels, by using two TCs
  - Two 24-bit and one 16-bit Timer/Counters for Control (TCC), with extended functions:
    - Up to four compare channels with optional complementary output
    - Generation of synchronized pulse width modulation (PWM) pattern across port pins
    - Deterministic fault protection, fast decay and configurable dead-time between complementary output
    - Dithering that increase resolution with up to 5-bit and reduce quantization error
  - 32-bit Real Time Counter (RTC) with clock/calendar function
  - Watchdog Timer (WDT)
  - CRC-32 generator
  - One full-speed (12 Mbps) Universal Serial Bus (USB) 2.0 interface
    - Embedded host and device function
    - Eight endpoints
  - Up to five Serial Communication Interfaces (SERCOM), each configurable to operate as either:
    - USART with full-duplex and single-wire half-duplex configuration
    - I^2C up to 3.4 MHz
    - SPI
    - LIN slave
  - One 12-bit, 1 MSPS Analog-to-Digital Converter (ADC) with up to eight external channels
    - Differential and single-ended input
    - Automatic offset and gain error compensation
    - Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution
  - Two Analog Comparators (AC) with window compare function
  - Peripheral Touch Controller (PTC)
    - 18-channel capacitive touch and proximity sensing
    - Wake-up on touch in standby mode
- I/O and Package

  - 16/28 programmable I/O pins
  - 32-pin and 48-pin QFN
- Operating Voltage
  - 1.8V – 3.6V
- Temperature Range
  - -40°C to 85°C Industrial
- Power Consumption
  - Transceiver with microcontroller in idle mode (TX output power +5 dBm):
    - RX_ON = 9.4 mA
    - BUSY_TX = 18.2 mA
  - Active mode for the microcontroller down to 60 µA/MHz
  - Standby mode for the microcontroller down to 1.4 µA/MHz

Table of Contents

1. Description

The SAM R30 is a series of Ultra low-power microcontrollers equipped with an IEEE® 802.15.4-2003/2006/2011-compliant RF interface for the sub-1 GHz frequency bands such as 780 MHz (China), 868 MHz (Europe) and 915 MHz (North America). It is using the 32-bit ARM® Cortex®-M0+ processor at max. 48 MHz (2.46 CoreMark®/MHz) and offers 256 KB of Flash and 40 KB of SRAM in both 32- and 48-pin packages. Sophisticated power management technologies, such as power domain gating, SleepWalking, Ultra low-power peripherals and more, allow for very low current consumptions.

The highly-configurable peripherals include a touch controller supporting capacitive interfaces with proximity sensing.

The sub-GHz RF interface supports OQPSK and BPSK formats per the IEEE specifications. Additional proprietary formats include high data-rate and wideband BPSK. Built-in features include spread spectrum radio, automated packet handing and power management. With link budgets up to 120 dBm, the SAMR30 transceiver is a great alternative to 2.4 GHz with power hungry range extenders.

The SAM R30 devices provide the following features: in-system programmable Flash, 16-channel direct memory access (DMA) controller, 12-channel Event System, programmable interrupt controller, up to 28 programmable I/O pins, 32-bit real-time clock and calendar, up to three 16-bit Timer/Counters (TC) and three Timer/Counters for Control (TCC), where each TC/TCC can be configured to perform frequency and waveform generation, accurate program execution timing or input capture with time and frequency measurement of digital signals. The TCs can operate in 8- or 16-bit mode, selected TCs can be cascaded to form a 32-bit TC, and three timer/counters have extended functions optimized for motor, lighting and other control applications. Two TCC can operate in 24-bit mode, the third TCC can operate in 16-bit mode. The series provide one full-speed USB 2.0 embedded host and device interface; up to six Serial Communication Modules (SERCOM) that each can be configured to act as an USART, UART, SPI, I^2C up to 3.4 MHz, SMBus, PMBus and LIN slave; up to twenty channel 1 MSPS 12-bit ADC with programmable gain and optional oversampling and decimation supporting up to 16-bit resolution, two analog comparators with window mode, Peripheral Touch Controller supporting up to 18 buttons, sliders, wheels and proximity sensing; programmable Watchdog Timer, brown-out detector and power-on reset and two-pin Serial Wire Debug (SWD) program and debug interface.

All devices have accurate low-power external and internal oscillators. All oscillators can be used as a source for the system clock. Different clock domains can be independently configured to run at different frequencies, enabling power saving by running each peripheral at its optimal clock frequency, thus maintaining a high CPU frequency while reducing power consumption.

The SAM R30 devices have three software-selectable sleep modes, idle, standby and backup. In idle mode, the CPU is stopped while all other functions may be kept running. In standby, all clocks and functions are stopped except those selected to continue running. In this mode, all RAMs and logic contents are retained. The device supports SleepWalking. This feature allows some peripherals to wake up from sleep based on predefined conditions, thus allowing some internal operations like DMA transfer and/or the CPU to wake up only when needed, for example, when a threshold is crossed or a result is ready. The Event System supports synchronous and asynchronous events, allowing peripherals to receive, react to and send events even in standby mode.

The SAM R30 devices have two software-selectable performance levels (PL0 and PL2) allowing the user to scale the lowest core voltage level that will support the operating frequency. To further minimize current consumption, specifically leakage dissipation, the SAM R30 devices utilize a power domain gating technique with retention to turn off some logic areas while keeping its logic state. This technique is fully handled in hardware.

The Flash program memory can be reprogrammed in-system through the SWD interface. The same interface can also be used for nonintrusive, on-chip debugging of application code. A boot loader running in the device can use any communication interface to download and upgrade the application program in the Flash memory.

The SAM R30 devices are supported with a full suite of programs and system development tools, including C compilers, macro assemblers, program debugger/simulators, programmers and evaluation kits.

2. Configuration Summary

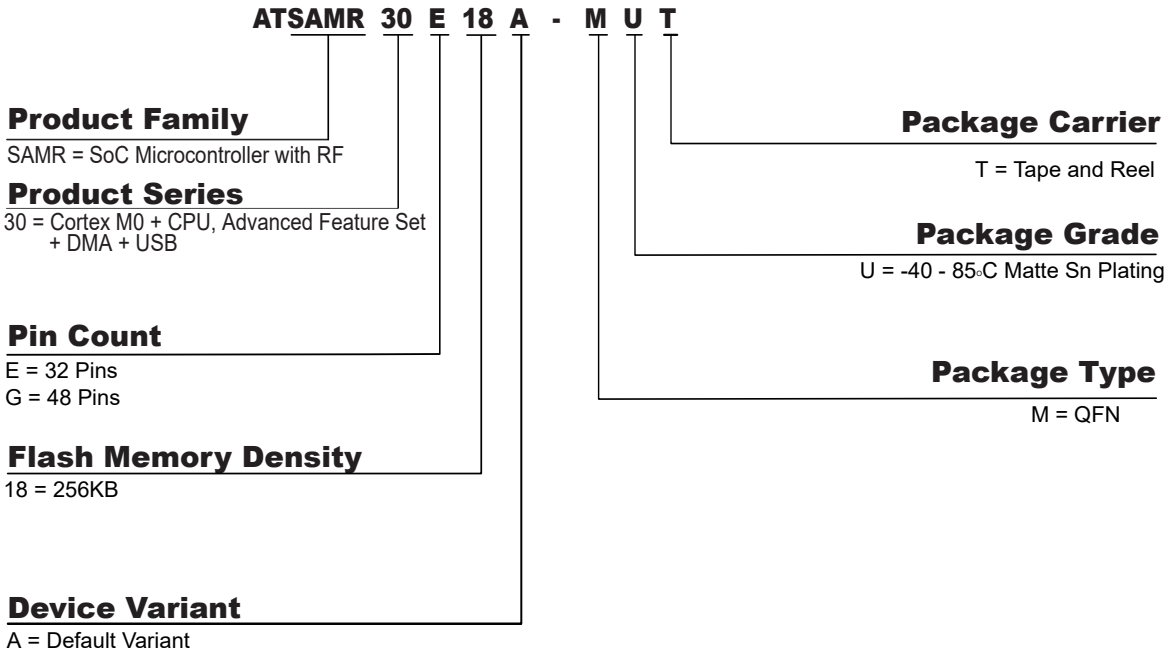
	SAM R30G	SAM R30E
Pins	48	32
General Purpose I/O-pins (GPIOs)	28	16
Flash	256 KB	256 KB
Flash RWW section	8 KB	8 KB
System SRAM	32 KB	32 KB
Low Power SRAM	8 KB	8 KB
Timer Counter (TC) instances	3	3
Waveform output channels per TC instance	2	2
Timer Counter for Control (TCC) instances	3	3
Waveform output channels per TCC	4/2/2	4/2/2
USB interface	1	1
Serial Communication Interface (SERCOM) instances	5+1 (1)	4+1 (1)
Inter-IC Sound (I²S) interface	No	No
Analog-to-Digital Converter (ADC) channels	8	4
Analog Comparators (AC)	2	2
Digital-to-Analog Converter (DAC) channels	No	No
Real-Time Counter (RTC)	Yes	Yes
RTC alarms	1	1
RTC compare values	1 32-bit value or 2 16-bit values	1 32-bit value or 2 16-bit values
External Interrupt lines	16	14
Peripheral Touch Controller (PTC) channels (X- x Y-Lines) for mutual capacitance (2)	6x3	4x3
Peripheral Touch Controller (PTC) channels for self capacitance (Y-Lines only) (3)	3	3
Maximum CPU frequency	48 MHz	
Packages	QFN	QFN
32.768 kHz crystal oscillator (XOSC32K)	Yes	No

.....continued

	SAM R30G	SAM R30E
Oscillators	16 MHz crystal oscillator for TRX (XOSCRF) 0.4-32 MHz crystal oscillator (XOSC) 32.768 kHz internal oscillator (OSC32K) 32 kHz ultra-low-power internal oscillator (OSCULP32K) 16/12/8/4 MHz high-accuracy internal oscillator (OSC16M) 48 MHz Digital Frequency Locked Loop (DFLL48M) 96 MHz Fractional Digital Phased Locked Loop (FDPLL96)	
Event System channels	12	12
SW Debug Interface	Yes	Yes
Watchdog Timer (WDT)	Yes	Yes

1. SERCOM4 is internally connected to the AT86RF212B.
2. The number of X- and Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines. Refer to Multiplexed Signals for details. The number in the Configuration Summary is the maximum number of channels that can be obtained.
3. The number of Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines. The number given here is the maximum number of Y-lines that can be obtained.

3. Ordering Information



3.1 SAM R30E

Table 3-1. SAM R30E

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAMR30E18A-MU	256K	32K	QFN32	Tray
ATSAMR30E18A-MUT	256K	32K	QFN32	Tape & Reel

3.2 SAM R30G

Table 3-2. SAM R30G

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAMR30G18A-MU	256K	32K	QFN48	Tray
ATSAMR30G18A-MUT	256K	32K	QFN48	Tape & Reel

3.3 Device Identification

The DSU - Device Service Unit peripheral provides the Device Selection bits in the Device Identification register (DID.DEVSEL) in order to identify the device by software. The SAM R30 variants have a reset value of DID=0x1081drxx, with the LSB identifying the die number ('d'), the die revision ('r') and the device selection ('xx').

Table 3-3. SAM R30 Device Identification Values

DEVSEL (DID[7:0])	Device
0x1081021E	SAM R30G18A
0x1081021F	SAM R30E18A

Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

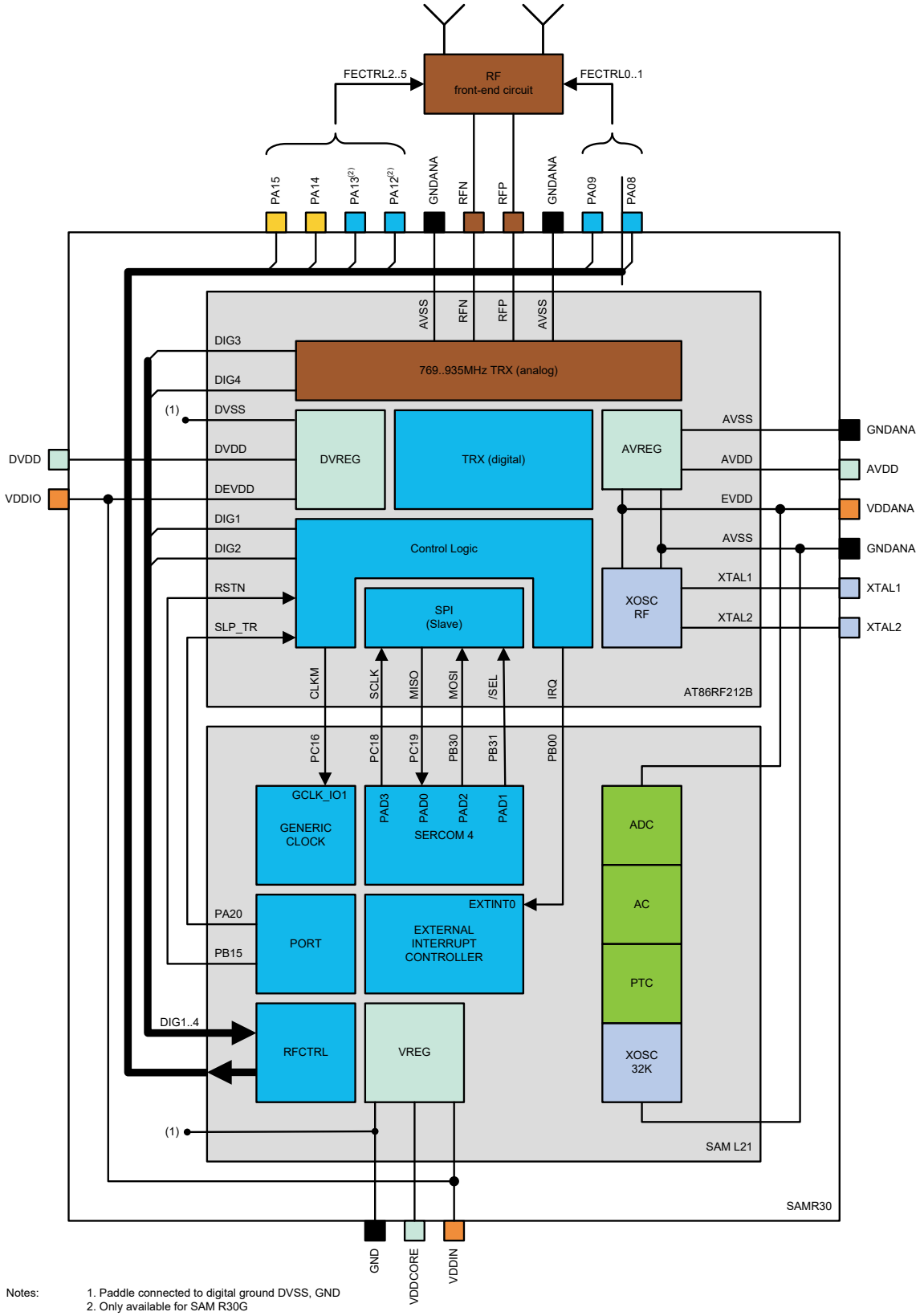
4. System Introduction

The SAM R30 SIP consists of two vertically integrated silicon dies:

- SAM L21 ARM® Cortex® M0+ based microcontroller.
- AT86RF212B low-power, low-voltage 700/800/900MHz transceiver

The local communication and control interface is wired within the package. Key I/O external signals are exposed as I/O pins. .

4.1 Interconnection Diagram

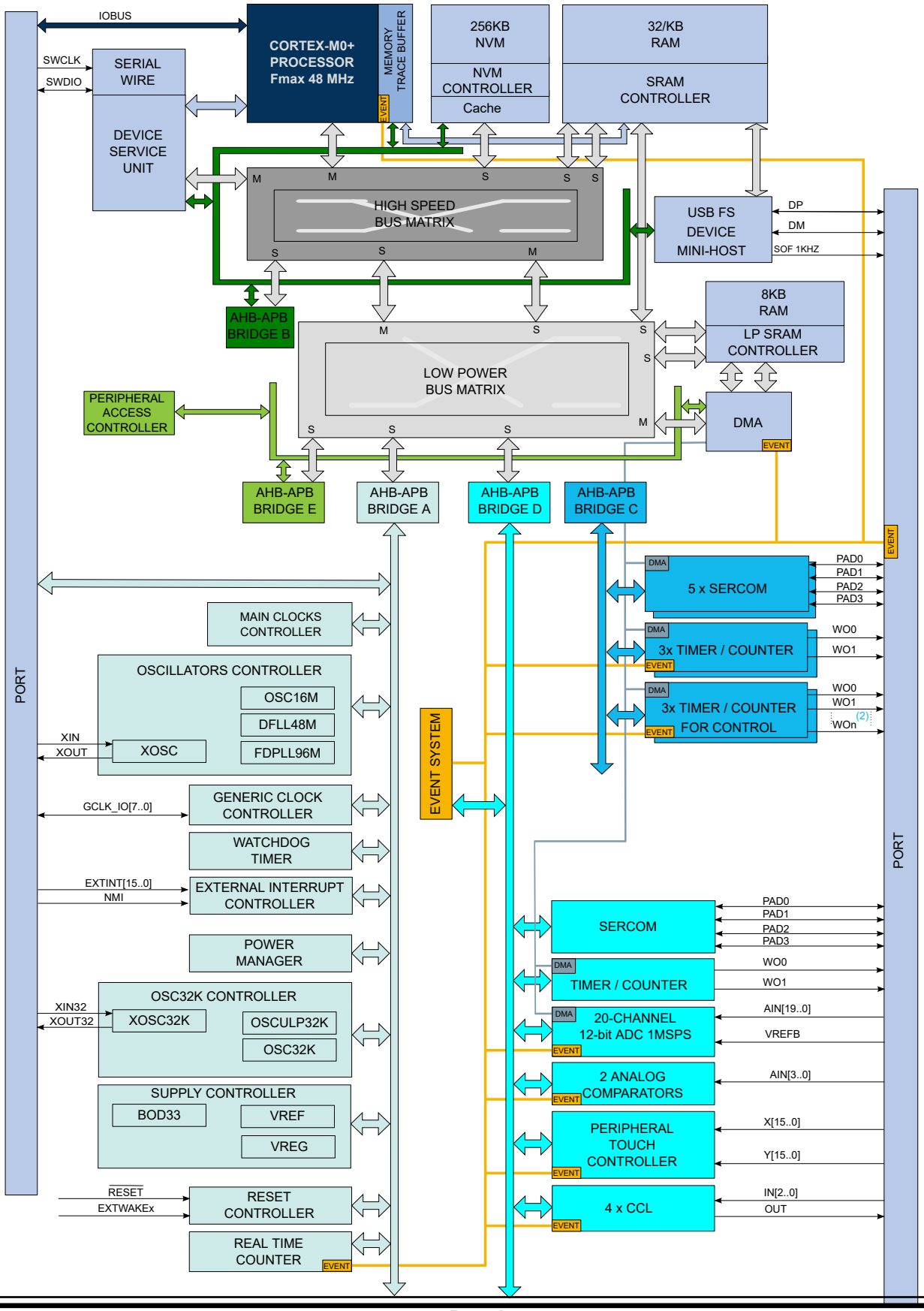


Notes:
1. Paddle connected to digital ground DVSS, GND
2. Only available for SAM R30G

Related Links

6.2 Inter-Die signal description
14.1.1 Overview
13.30.3 Block Diagram

4.2 MCU Block Diagram



Notes:

1. Some products have different number of SERCOM instances, Timer/Counter instances, PTC signals and ADC signals.
2. The three TCC instances have different configurations, including the number of Waveform Output (WO) lines.

Related Links

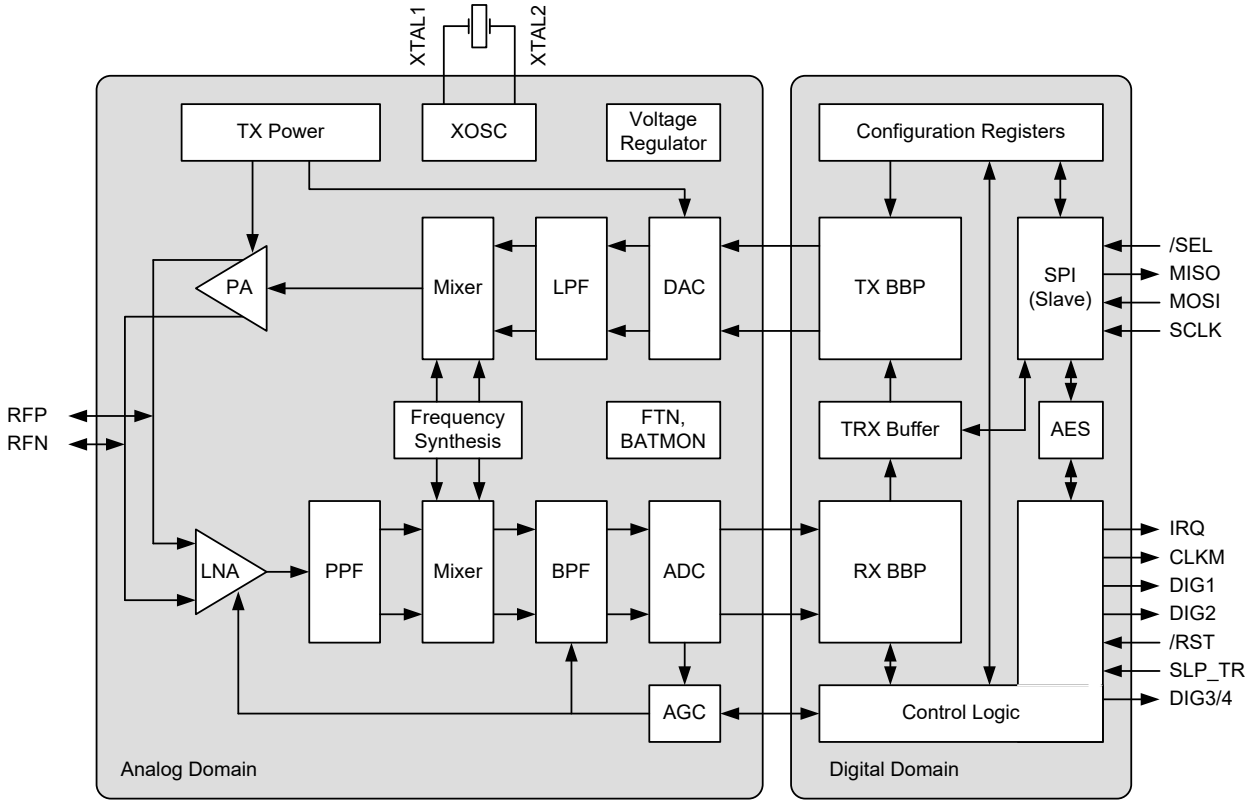
2. Configuration Summary

7.3.4 TCC Configurations

4.3 Transceiver Circuit Description

The AT86RF212B single-chip radio transceiver provides a complete radio transceiver interface between radio frequency signals and baseband microcontroller. It comprises a bidirectional analog RF front end, direct-conversion mixers, low-noise fractional-n PLL, quadrature digitizer, DSP modem and baseband packet-handler optimized for IEEE 802.15.4 MAC/PHY automation and low-power. An SPI accessible 128-byte TRX buffer stores receive or transmit data. Radio communication between transmitter and receiver is based on DSSS Spread Spectrum with OQPSK or BPSK modulation schemes as defined by the IEEE 802.15.4 standard. Additional proprietary modulation modes include high-data rate payload encoding and wideband BPSK-40-ALT.

Figure 4-1. AT86RF212B Block Diagram



The number of required external components is minimal. The basic requirements are an antenna, a balun, harmonic filter, local oscillator and bypass capacitors. The RF Ports are bidirectional 100Ω differential signals that do not require external TX/RX switches. Hardware control signals are automatically generated for TX/RX arbitration of high-powered PA/LNA frontends and transmitter diversity for systems with dual antennas.

The AT86RF212B supports the IEEE 802.15.4-2006 [2] standard mandatory BPSK modulation and optional O-QPSK modulation in the 868.3MHz and 915MHz bands. In addition, it supports the O-QPSK modulation defined in IEEE 802.15.4-2011 [4] for the Chinese 780MHz band. For applications not targeting IEEE compliant networks, the radio transceiver supports proprietary High Data Rate Modes based on O-QPSK. Additionally the AT86RF212B provides BPSK-40-ALT wideband BPSK mode for compliance with FCC rule 15.247 and backward compatibility with legacy BPSK networks.

The AT86RF212B features hardware supported 128-bit security operation. The standalone AES encryption/ decryption engine can be accessed in parallel to all PHY operational modes. Configuration of the AT86RF212B, reading and writing of data memory, as well as the AES hardware engine are controlled by the SPI interface and additional control signals.

On-chip low-dropout linear regulators provide clean 1.8 V_{DC} power for critical analog and digital sub-systems. To conserve power, these rails are automatically sequenced by the transceiver's state machine. This feature greatly improves EMC in the RF domain and reduces external power supply complexity to the simple addition of frequency compensation capacitors on the AVDD and DVDD pins.

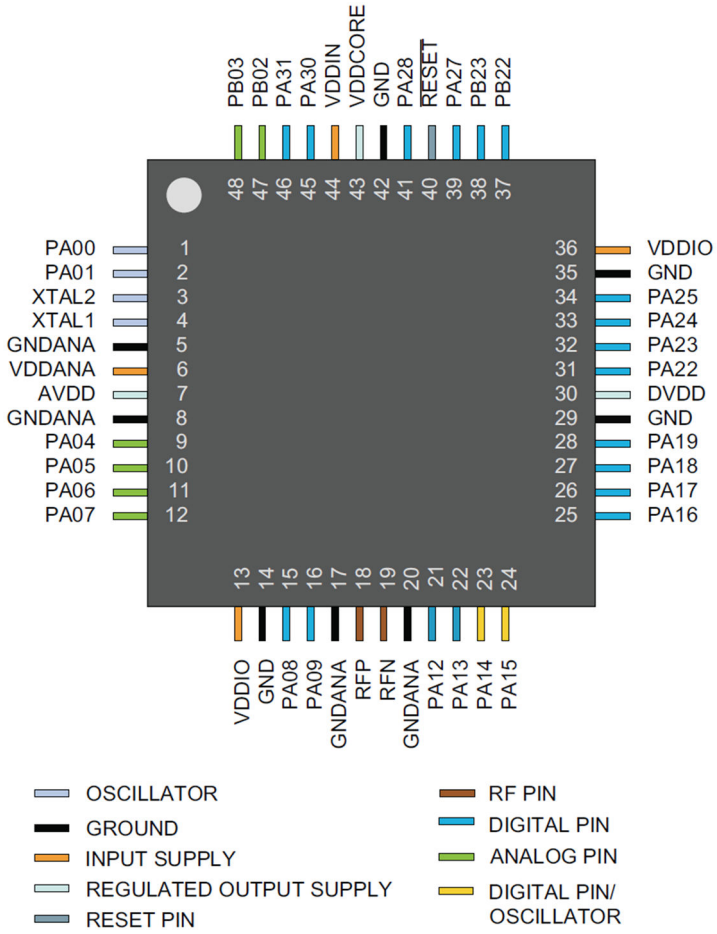
Additional features of the Extended Feature Set are provided to simplify the interaction between radio transceiver and microcontroller.

Related Links

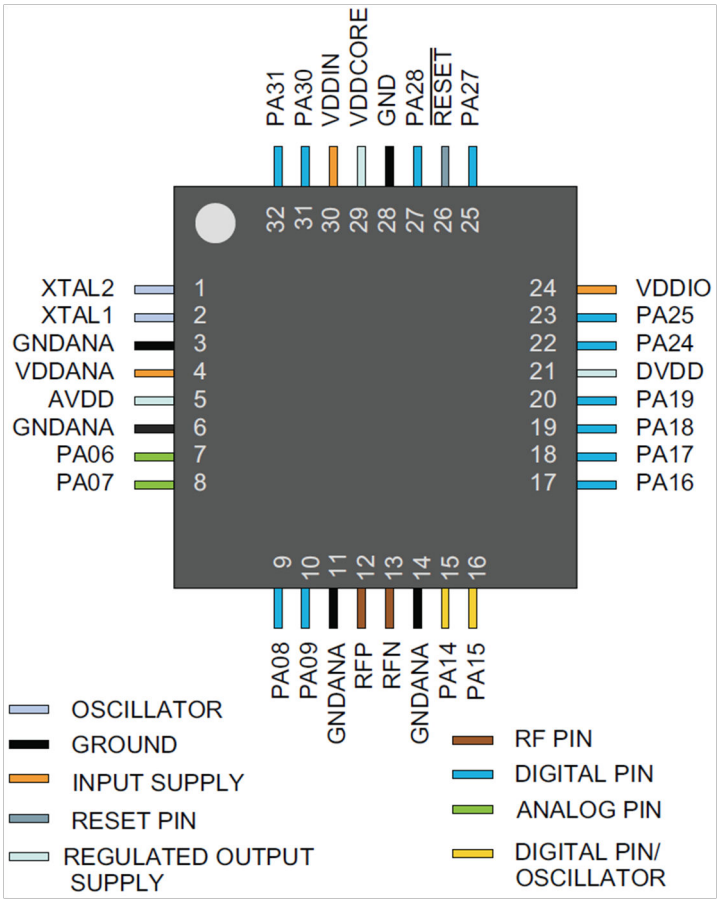
23. References

5. Pinout

5.1 SAM R30G



5.2 SAM R30E


PA31
PA30
VDDIN
VDDCORE
GND
PA28
RESET
PA27
32
31
30
29
28
27
26
25
XTAL2 1
XTAL1 2
GNDANA 3
VDDANA 4
AVDD 5
GNDANA 6
PA06 7
PA07 8
24 VDDIO
23 PA25
22 PA24
21 DVDD
20 PA19
19 PA18
18 PA17
17 PA16
9
10
11
12
13
14
15
16
PA08
PA09
GNDANA
RFP
RFN
GNDANA
PA14
PA15
OSCILLATOR
GROUND
INPUT SUPPLY
RESET PIN
REGULATED OUTPUT SUPPLY
RF PIN
DIGITAL PIN
ANALOG PIN
DIGITAL PIN/ OSCILLATOR

6. Signal Description

Functional description of signals available at the package or routed in between the system dies.

The nature of a SIP results in the situation where the package pins may be bonded to the microcontroller die or the transceiver die. There are also signals bonded in between the two dies. This section will provide the required information to understand the origin and the function of each such signal.

6.1 Signal Description

The following table gives details on signal names classified by peripheral.

Table 6-1. Signal Descriptions List

Signal Name	Function	Type	Active Level
Analog Comparators – AC			
AIN[3:0]	AC Analog Inputs	Analog	—
CMP[1:0]	AC Comparator Outputs	Digital	—
Analog Digital Converter – ADC			
AIN[19:0]	ADC Analog Inputs	Analog	—
VREFB	ADC Voltage External Reference B	Analog	—
External Interrupt Controller – EIC			
EXTINT[15:0]	External Interrupts inputs	Digital	—
NMI	External Non-Maskable Interrupt input	Digital	—
Reset Controller – RSTC			
EXTWAKE[7:0]	External wake-up inputs	Digital	—
Generic Clock Generator – GCLK			
GCLK_IO[7:0]	Generic Clock (source clock inputs or generic clock generator output)	Digital	—
Custom Control Logic – CCL			
IN[11:0]	Logic Inputs	Digital	—
OUT[3:0]	Logic Outputs	Digital	—
Supply Controller – SUPC			
VBAT	External battery supply Inputs	Analog	—
PSOK	Main Power Supply OK input	Digital	—
OUT[1:0]	Logic Outputs	Digital	—
Power Manager – PM			
RESETN	Reset input	Digital	Low
Serial Communication Interface – SERCOMx			
PAD[3:0]	SERCOM Inputs/Outputs Pads	Digital	—
Oscillators Control – OSCCTRL			

.....continued			
Signal Name	**Function**	**Type**	**Active Level**
XIN	Crystal or external clock Input	Analog/Digital	—
XOUT	Crystal Output	Analog	—
32KHz Oscillators Control – OSC32KCTRL			
XIN32	32 KHz Crystal or external clock Input	Analog/Digital	—
XOUT32	32 KHz Crystal Output	Analog	—
Timer Counter – TCx			
WO[1:0]	Waveform Outputs	Digital	—
Timer Counter – TCCx			
WO[7:0]	Waveform Outputs	Digital	—
Peripheral Touch Controller – PTC			
X[15:0]	PTC Input	Analog	—
Y[15:0]	PTC Input	Analog	—
General Purpose I/O – PORT			
PA01 – PA00	Parallel I/O Controller I/O Port A	Digital	—
PA09 – PA04	Parallel I/O Controller I/O Port A	Digital	—
PA19 – PA12	Parallel I/O Controller I/O Port A	Digital	—
PA25 – PA22	Parallel I/O Controller I/O Port A	Digital	—
PA28 – PA27	Parallel I/O Controller I/O Port A	Digital	—
PA31–PA30	Parallel I/O Controller I/O Port A	Digital	—
PB03 – PB02	Parallel I/O Controller I/O Port B	Digital	—
PB23 – PB22	Parallel I/O Controller I/O Port B	Digital	—
Universal Serial Bus – USB			
DP	DP for USB	Digital	—
DM	DM for USB	Digital	—
SOF 1 kHz	USB Start of Frame	Digital	—

6.2 Inter-Die signal description

Table 6-2. RF212B signals bonded inside the SIP

Signal Name	Function	Type	Signal is connected to
DIG3	1. RX/TX Indication 2. If disabled, pull-down enabled (AVSS)	Digital output	RFCTRL

.....continued

Signal Name	Function	Type	Signal is connected to
DIG4	1. RX/TX Indication (DIG3 inverted) 2. If disabled, pull-down enabled (AVSS)	Digital output	RFCTRL
/RST	Chip reset; active low	Digital input	L21 PB15
DIG1	1. Antenna Diversity RF switch control 2. If disabled, pull-down enabled (DVSS)	Digital output	RFCTRL
DIG2	1. Antenna Diversity RF switch control (DIG1 inverted) 2. RX Frame Time Stamping 3. If functions disabled, pull-down enabled (DVSS)	Digital output	RFCTRL
SLP_TR	Controls sleep, transmit start, and receive states; active high	Digital input	L21 PA20
CLKM	Master clock signal output; low if disabled	Digital output	L21 PC16
SCLK	SPI clock	Digital input	L21 PC18
MISO	SPI data output (master input slave output)	Digital output	L21 PC19
MOSI	SPI data input (master output slave input)	Digital input	L21 PB30
/SEL	SPI select, active low	Digital input	L21 PB31
IRQ	1. Interrupt request signal; active high or active low; configurable 2. Frame Buffer Empty Indicator; active high	Digital output	L21 PB00

6.3 AT86RF212B Pin Description

Table 6-3. Description of AT86RF212B signals available outside the package.

Name	Type	Description
RFP	RF I/O	Differential RF signal
RFN	RF I/O	Differential RF signal
AVDD	Supply	Frequency compensation connection for internal 1.8 V_{DC} analog power supply.
DVDD	Supply	Frequency compensation connection for internal 1.8 V_{DC} digital power supply.

.....continued

Name	Type	Description
DEVDD	Supply	External supply voltage; digital domain
EVDD	Supply	External supply voltage, analog domain
XTAL2	Analog output	Crystal pin
XTAL1	Analog input	Crystal pin or external clock supply
DVSS	Ground	Digital ground
AVSS	Ground	Analog ground or Ground for RF signals

7. I/O Multiplexing and Considerations

7.1 Multiplexed Signals

Each pin is controlled by the PORT as a general purpose I/O by default and, alternatively, can be assigned to one of the peripheral functions A, B, C, D, E, F, G, H or I. To enable a peripheral function on a pin, the Peripheral Multiplexer Enable bit in the Pin Configuration register corresponding to that pin (PINCFGn.PMUXEN, n = 0..31) in the PORT must be written to '1'. The selection of peripheral function A to H is done by writing to the Peripheral Multiplexing Odd and Even bits in the Peripheral Multiplexing register (PMUXn.PMUXE/O) in the PORT.

This table describes the peripheral signals multiplexed to the PORT I/O pins.

Table 7-1. Port Function Multiplexing

PIN		I/O Pin	Supply	A		B(1)(2)					C	D	E	F	G	H	I
SAMR30E	SAMR30G			EIC	RSTC	AC	ADC	REF	PTC X-lines	PTC Y-lines	SERCOM(1)(2)	SERCOM-ALT	TC/ TCC	FECTRL/TCC	COM	AC/ GCLK/ SUPC	CCL
—	1	PA00	VSWOUT	EXTINT[0]	EXTWAKE[0]	—	—	—	—	—	—	SERCOM1/ PAD[0]	TCC2/ WO[0]	—	—	—	—
—	2	PA01	VSWOUT	EXTINT[1]	EXTWAKE[1]	—	—	—	—	—	—	SERCOM1/ PAD[1]	TCC2/ WO[1]	—	—	—	—
—	9	PA04	VDDANA	EXTINT[4]	EXTWAKE[4]	AIN[0]	AIN[4]	ADC/ VREFB	—	—	—	SERCOM0/ PAD[0]	TCC0/ WO[0]	—	—	—	CCL0/IN[0]
—	10	PA05	VDDANA	EXTINT[5]	EXTWAKE[5]	AIN[1]	AIN[5]	—	—	—	—	SERCOM0/ PAD[1]	TCC0/ WO[1]	—	—	—	CCL0/IN[1]
7	11	PA06	VDDANA	EXTINT[6]	EXTWAKE[6]	AIN[2]	AIN[6]	—	—	Y[4]	—	SERCOM0/ PAD[2]	TCC1/ WO[0]	—	—	—	CCL0/IN[2]
8	12	PA07	VDDANA	EXTINT[7]	EXTWAKE[7]	AIN[3]	AIN[7]	—	—	—	—	SERCOM0/ PAD[3]	TCC1/ WO[1]	—	—	—	CCL0/OUT
9	15	PA08	VDDIO	NMI	—	—	AIN[16]	—	X[0]	Y[6]	SERCOM0/ PAD[0]	SERCOM2/ PAD[0]	TCC0/ WO[0]	FECTRL[0]	—	—	CCL1/IN[0]
10	16	PA09	VDDIO	EXTINT[9]	—	—	AIN[17]	—	X[1]	Y[7]	SERCOM0/ PAD[1]	SERCOM2/ PAD[1]	TCC0/ WO[1]	FECTRL[1]	—	—	CCL1/IN[1]
—	21	PA12	VDDIO	EXTINT[12]	—	—	—	—	—	—	SERCOM2/ PAD[0]	SERCOM4/ PAD[0]	TCC2/ WO[0]	FECTRL[2]	—	AC/CMP[0]	—
—	22	PA13	VDDIO	EXTINT[13]	—	—	—	—	—	—	SERCOM2/ PAD[1]	SERCOM4/ PAD[1]	TCC2/ WO[1]	FECTRL[3]	—	AC/CMP[1]	—
15	23	PA14	VDDIO	EXTINT[14]	—	—	—	—	—	—	SERCOM2/ PAD[2]	SERCOM4/ PAD[2]	TC4/WO[0]	FECTRL[4]	—	GCLK/IO[0]	—
16	24	PA15	VDDIO	EXTINT[15]	—	—	—	—	—	—	SERCOM2/ PAD[3]	SERCOM4/ PAD[3]	TC4/WO[1]	FECTRL[5]	—	GCLK/IO[1]	—
17	25	PA16	VDDIO	EXTINT[0]	—	—	—	—	X[4]	—	SERCOM1/ PAD[0]	SERCOM3/ PAD[0]	TCC2/ WO[0]	TCC0/WO[6]	—	GCLK/IO[2]	CCL0/IN[0]
18	26	PA17	VDDIO	EXTINT[1]	—	—	—	—	X[5]	—	SERCOM1/ PAD[1]	SERCOM3/ PAD[1]	TCC2/ WO[1]	TCC0/WO[7]	—	GCLK/IO[3]	CCL0/IN[1]
19	27	PA18	VDDIO	EXTINT[2]	—	—	—	—	X[6]	—	SERCOM1/ PAD[2]	SERCOM3/ PAD[2]	TC4/WO[0]	TCC0/WO[2]	—	AC/CMP[0]	CCL0/IN[2]
20	28	PA19	VDDIO	EXTINT[3]	—	—	—	—	X[7]	—	SERCOM1/ PAD[3]	SERCOM3/ PAD[3]	TC4/WO[1]	TCC0/WO[3]	—	AC/CMP[1]	CCL0/OUT
—	31	PA22	VDDIO	EXTINT[6]	—	—	—	—	X[10]	—	SERCOM3/ PAD[0]	SERCOM5/ PAD[0]	TC0/WO[0]	TCC0/WO[4]	—	GCLK/IO[6]	CCL2/IN[0]
—	32	PA23	VDDIO	EXTINT[7]	—	—	—	—	X[11]	—	SERCOM3/ PAD[1]	SERCOM5/ PAD[1]	TC0/WO[1]	TCC0/WO[5]	USB/ SOF_1KHZ	GCLK/IO[7]	CCL2/IN[1]
22	33	PA24	VDDIO	EXTINT[12]	—	—	—	—	—	—	SERCOM3/ PAD[2]	SERCOM5/ PAD[2]	TC1/WO[0]	TCC1/WO[2]	USB/DM	—	CCL2/IN[2]
23	34	PA25	VDDIO	EXTINT[13]	—	—	—	—	—	—	SERCOM3/ PAD[3]	SERCOM5/ PAD[3]	TC1/WO[1]	TCC1/WO[3]	USB/DP	—	CCL2/OUT
—	37	PB22	VDDIN	EXTINT[6]	—	—	—	—	—	—	—	SERCOM5/ PAD[2]	—	—	—	GCLK/IO[0]	CCL0/IN[0]
—	38	PB23	VDDIN	EXTINT[7]	—	—	—	—	—	—	—	SERCOM5/ PAD[3]	—	—	—	GCLK/IO[1]	CCL0/OUT

.....continued

PIN		I/O Pin	Supply	A		B[1][2]					C	D	E	F	G	H	I
SAMR30E	SAMR30G			EIC	RSTC	AC	ADC	REF	PTC X-lines	PTC Y-lines	SERCOM[1][2]	SERCOM-ALT	TC/ TCC	FECTRL/TCC	COM	AC/ GCLK/ SUPC	CCL
25	39	PA27	VDDIN	EXTINT[15]	—	—	—	—	—	—	—	—	—	—		GCLK/IO[0]	—
27	41	PA28	VDDIN	EXTINT[8]	—	—	—	—	—	—	—	—	—	—	—	GCLK/IO[0]	—
31	45	PA30	VDDIN	EXTINT[10]	—	—	—	—	—	—	—	SERCOM1/ PAD[2]	TCC1/ WO[0]	—	CM0P/SWCLK	GCLK/IO[0]	CCL1/IN[0]
32	46	PA31	VDDIN	EXTINT[11]	—	—	—	—	—	—	—	SERCOM1/ PAD[3]	TCC1/ WO[1]	—	SWDIO[3]	—	CCL1/OUT
—	47	PB02	VSWOUT	EXTINT[2]	—	—	AIN[10]	—	—	—	—	SERCOM5/ PAD[0]	—	—	—	SUPC/ OUT[1]	CCL0/OUT
—	48	PB03	VSWOUT	EXTINT[3]	—	—	AIN[11]	—	—	—	—	SERCOM5/ PAD[1]	—	—	—	SUPC/VBAT	—

1. All analog pin functions are on peripheral function B. Peripheral function B must be selected to disable the digital control of the pin.
2. Only some pins can be used in SERCOM I^2C mode. See also 7.3.3 SERCOM I2C Pins.
3. This function is only activated in the presence of a debugger.
4. When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of this pin. This is also true even when the peripheral is used for internal purposes.
5. Clusters of multiple GPIO pins are sharing the same supply pin.

7.2 Internal Multiplexed Signals

PA20, PB00, PB15, PB30, PB31, PC16, PC18 and PC19 are controlled by the PORT as general purpose I/O by default and, alternatively, may be assigned to one of the peripheral functions A, B, C, D, E, F, G or H. To enable a peripheral function on a pin, the Peripheral Multiplexer Enable bit in the Pin Configuration register corresponding to that pin (PINCFGn.PMUXEN, n = 0-31) in the PORT must be written to one. The selection of peripheral functions A to H is done by writing to the Peripheral Multiplexing Odd and Even bits in the Peripheral Multiplexing register (PMUXn.PMUXE/O) in the PORT.

PA10, PA11, PB16 and PB17 cannot be configured as output ports. These ports are always connected to the RFCTRL inputs.

Table 7-2. Internal Multiplexed Signals

				A	A	B	B	B	B	B	C	D	E	F	G	H	I
Internal Signal	IO Pin	Supply	Type	EIC	RSTC	REF	ADC	AC	PTC X-lines	PTC Y-lines	SERCOM	SERCOM-ALT	TC/ TCC	FECTRL/ TCC/ SERCOM	COM	AC/ GCLK	CCL
DIG3	PA10	VDDIO	Input	EXTINT[10]	—	—	AIN[18]	—	X[2]	Y[8]	SERCOM0/PAD[2]	SERCOM2/PAD[2]	TCC1/WO[0]	TCC0/WO[2]	—	GCLK_IO[4]	CCL1/IN[5]
DIG4	PA11	VDDIO	Input	EXTINT[11]	—	—	AIN[19]	—	X[3]	Y[9]	SERCOM0/PAD[3]	SERCOM2/PAD[3]	TCC1/WO[1]	TCC0/WO[3]	—	GCLK_IO[5]	CCL1/OUT[1]
SLP_TR	PA20	VDDIO	I/O	EXTINT[4]	—	—	—	—	X[8]	—	SERCOM5/PAD[2]	SERCOM3/PAD[2]	TC3/WO[0]	TCC0/WO[6]	—	GCLK_IO[4]	—
IRQ	PB00	VDDANA	I/O	EXTINT[0]	—	—	AIN[8]	—	—	—	—	SERCOM5/PAD[2]	TC3/WO[0]	—	—	SUPC/PSOK	CCL0/IN[1]
RSTN	PB15	VDDIO	I/O	EXTINT[15]	—	—	—	—	X[15]	—	SERCOM4/PAD[3]	—	TC1/WO[1]	—	—	GCLK_IO[1]	CCL3/IN[10]
DIG1	PB16	VDDIO	Input	EXTINT[0]	—	—	—	—	—	—	SERCOM5/PAD[0]	—	TC2/WO[0]	TCC0/WO[4]	—	GCLK_IO[2]	CCL3/IN[11]
DIG2	PB17	VDDIO	Input	EXTINT[1]	—	—	—	—	—	—	SERCOM5/PAD[1]	—	TC2/WO[1]	TCC0/WO[5]	—	GCLK_IO[3]	CCL3/OUT[3]
MOSI	PB30	VDDIO	I/O	EXTINT[14]	—	—	—	—	—	—	—	SERCOM5/PAD[0]	TCC0/WO[0]	SERCOM4/PAD[2]	—	—	—
SEL	PB31	VDDIO	I/O	EXTINT[15]	—	—	—	—	—	—	—	SERCOM5/PAD[1]	TCC0/WO[1]	SERCOM4/PAD[1]	—	—	—
CLKM	PC16	VDDIO	I/O	—	—	—	—	—	—	—	—	—	—	—	—	GCLK_IO[1]	—
SCLK	PC18	VDDIO	I/O	—	—	—	—	—	—	—	—	—	—	SERCOM4/PAD[3]	—	—	—
MISO	PC19	VDDIO	I/O	—	—	—	—	—	—	—	—	—	—	SERCOM4/PAD[0]	—	—	—

7.3 Other Functions

7.3.1 Oscillator Pinout

The oscillators are not mapped to the normal PORT functions and their multiplexing are controlled by registers in the Oscillators Controller (OSCCTRL) and in the 32KHz Oscillators Controller (OSC32KCTRL).

Table 7-3. Oscillator Pinout

Oscillator	Supply	Signal	I/O pin
XOSC	VDDIO	XIN	PA14
		XOUT	PA15
XOSC32K	VSWOUT	XIN32	PA00
		XOUT32	PA01

Note: To improve the cycle-to-cycle jitter of XOSC32, it is recommended to keep the neighboring pins of XIN32 and XOUT32 following pins as static as possible.

Table 7-4. XOSC32K Jitter Minimization

Package Pin Count	Static Signal Recommended
48	PB02, PB03, PA02, PA03

7.3.2 Serial Wire Debug Interface Pinout

Only the SWCLK pin is mapped to the normal PORT functions. A debugger cold-plugging or hot-plugging detection will automatically switch the SWDIO port to the SWDIO function.

Table 7-5. Serial Wire Debug Interface Pinout

Signal	Supply	I/O pin
SWCLK	VDDIN	PA30
SWDIO	VDDIN	PA31

7.3.3 SERCOM I^2C Pins

Table 7-6. SERCOM Pins Supporting I^2C

Device	Pins Supporting I^2C Hs mode
SAMR30E	PA08, PA09, PA16, PA17
SAMR30G	PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23

7.3.4 TCC Configurations

The SAM R30 has three instances of the Timer/Counter for Control applications (TCC) peripheral, , TCC[2:0]. The following table lists the features for each TCC instance.

Table 7-7. TCC Configuration Summary

TCC#	Channels (CC_NUM)	Waveform Output (WO_NUM)	Counter size	Fault	Dithering	Output matrix	Dead Time Insertion (DTI)	SWAP	Pattern generation
0	4	8	24-bit	Yes	Yes	Yes	Yes	Yes	Yes
1	2	4	24-bit	Yes	Yes				Yes

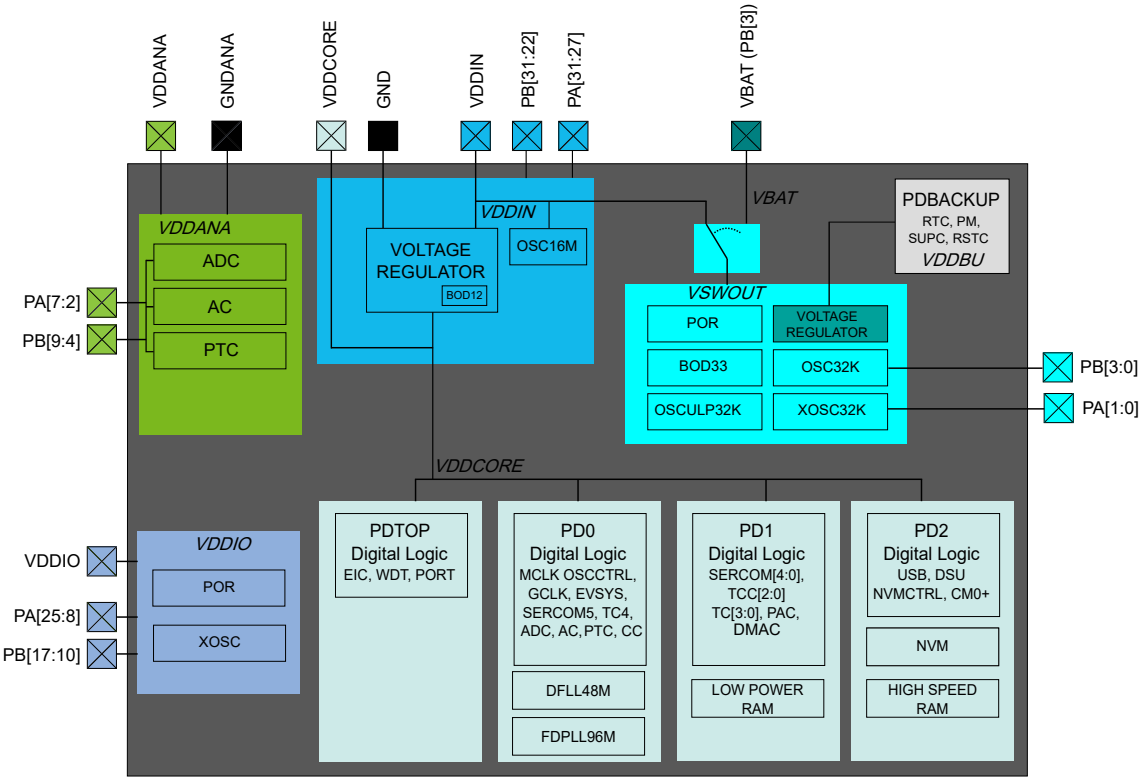
.....continued

TCC#	Channels (CC_NUM)	Waveform Output (WO_NUM)	Counter size	Fault	Dithering	Output matrix	Dead Time Insertion (DTI)	SWAP	Pattern generation
2	2	2	16-bit	Yes					

Note: The number of CC registers (CC_NUM) for each TCC corresponds to the number of compare/capture channels, so that a TCC can have more Waveform Outputs (WO_NUM) than CC registers.

8. Power Supply and Start-Up Considerations

8.1 Power Domain Overview



The SAM R30 power domains operate independently of each other:

- VDDCORE, VDDIO and VDDIN share GND, whereas VDDANA refers to GNDANA.
- VDDANA and VDDIN must share the main supply, VDD.
- VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals, DFLL48M and FDPLL96M.
- VSWOUT and VDDBU are internal power domains.
- On SAM L21E, VDDIO is electrically connected to the VDDANA domain and supplied through VDDANA.

8.2 Power Supply Considerations

8.2.1 Power Supplies

The SAM R30 has several different power supply pins:

- VDDIO powers I/O lines and XOSC. Voltage is 1.8V to 3.63V
- VDDIN powers I/O lines, OSC16M, the internal regulator for VDDCORE and the Automatic Power Switch. Voltage is 1.8V to 3.63V
- VDDANA powers I/O lines and the ADC, AC, and PTC. Voltage is 1.8V to 3.63V
- VBAT powers the Automatic Power Switch. Voltage is 1.8V to 3.63V
- VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals, DFLL48M and FDPLL96M. Voltage is 0.9V to 1.2V typically.
- The Automatic Power Switch is a configurable switch that selects between VDDIN and VBAT as the supply for the internal output VSWOUT; see the figure in 8.1 Power Domain Overview.

The same voltage must be applied to both VDDIN and VDDANA. This common voltage is referred to as VDD in the data sheet.

When the Peripheral Touch Controller (PTC) is used, VDDIO must be equal to VDD. When the PTC is not used by the user application, VDDIO may be lower than VDD.

The ground pins, GND, are common to VDDCORE, VDDIO and VDDIN. The ground pin for VDDANA is GNDANA.

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

8.2.2 Voltage Regulator

The SAM R30 internal Voltage Regulator has four different modes:

- Linear mode: This is the default mode when CPU and peripherals are running. It does not require an external inductor.
- Switching mode. This is the most efficient mode when the CPU and peripherals are running. This mode can be selected by software on the fly.
- Low Power (LP) mode. This is the default mode used when the chip is in standby mode.
- Shutdown mode. When the chip is in backup mode, the internal regulator is off.

Note that the Voltage Regulator modes are controlled by the Power Manager.

8.2.3 Typical Powering Schematic

The SAM R30 uses a single supply from 1.8V to 3.63V.

The following figure shows the recommended power supply connection.

Figure 8-1. Power Supply Connection for Linear Mode

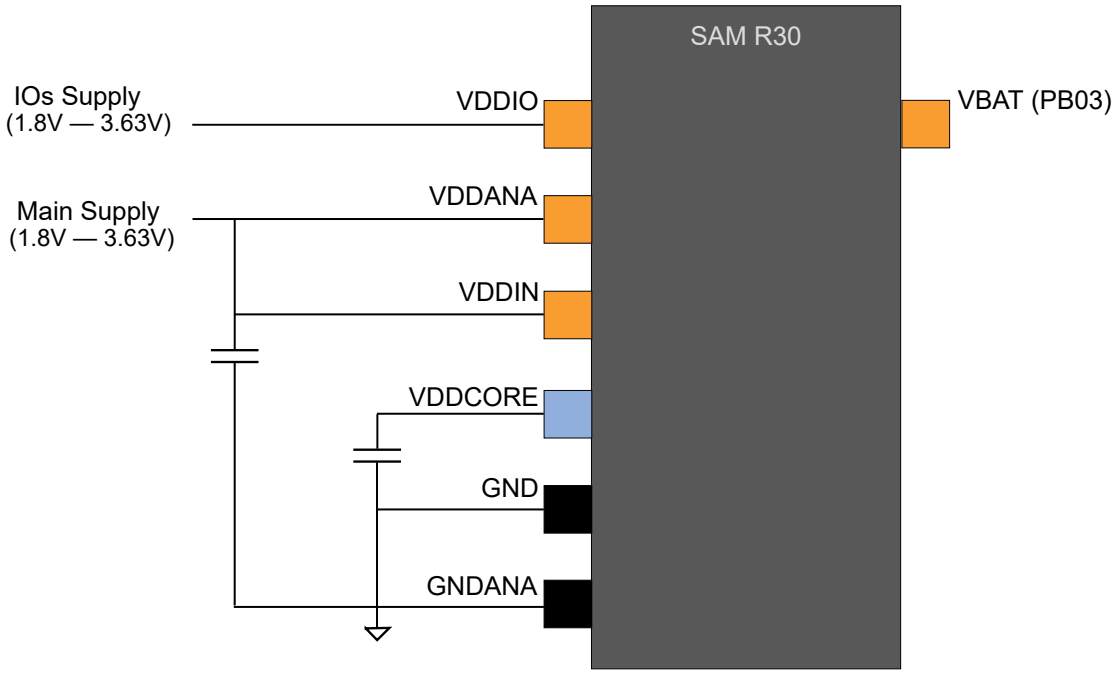
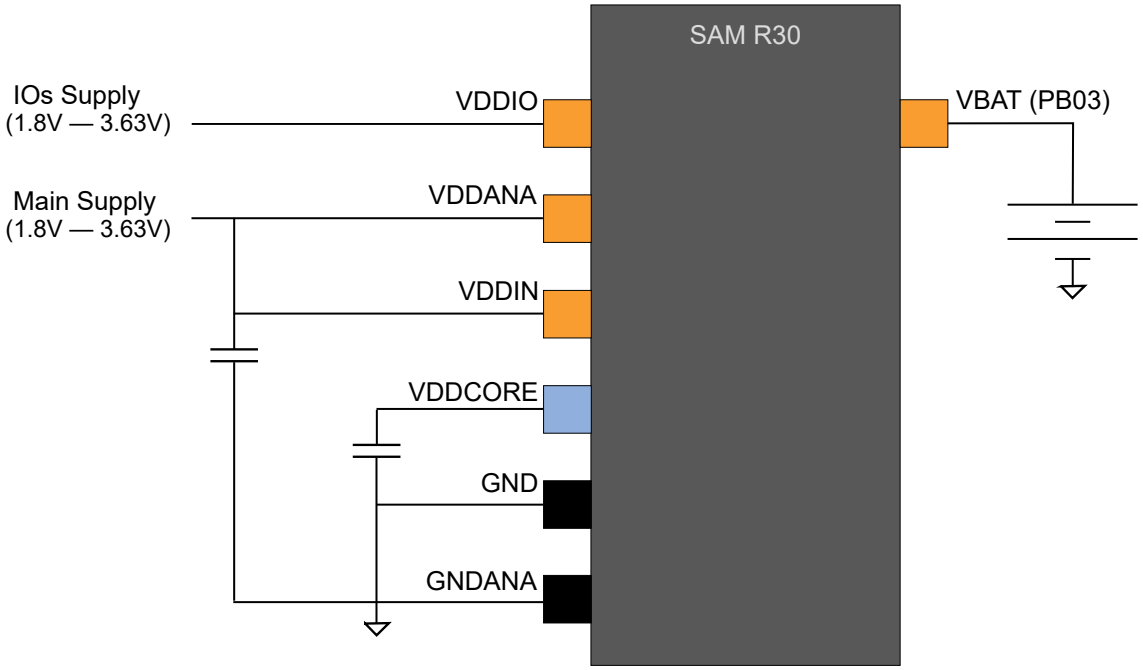


Figure 8-2. Power Supply Connection for Battery Backup



8.2.4 Power-Up Sequence

8.2.4.1 Supply Order

VDDIN and VDDANA must have the same supply sequence. Ideally, they must be connected together.

VDDIO can rise before or after VDDIN and VDDANA. Note that VDDIO supplies the XOSC, so VDDIO must be present before the application uses the XOSC feature. This is also applicable to all digital features present on pins supplied by VDDIO.

8.2.4.2 Minimum Rise Rate

The two integrated power-on reset (POR) circuits monitoring VDDIN and VDDIO require a minimum rise rate.

Related Links

15. Electrical Characteristics

8.2.4.3 Maximum Rise Rate

The rise rate of the power supplies must not exceed the values described in Electrical Characteristics.

Related Links

15. Electrical Characteristics

8.3 Power-Up

This section summarizes the power-up sequence of the SAM R30. The behavior after power-up is controlled by the Power Manager.

Related Links

13.8 PM – Power Manager

8.3.1 Starting of Internal Regulator

After power-up, the device is set to its initial state and kept in Reset, until the power has stabilized throughout the device. The default performance level after power-up is PL0. See section on *PM-Power Manager* for details. The

internal regulator provides the internal VDDCORE corresponding to this performance level. Once the external voltage VDDIN and the internal VDDCORE reach a stable value, the internal Reset is released.

Related Links

13.8 PM – Power Manager

8.3.2 Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 4MHz clock by default. The clock source for this clock signal is OSC16M, which is enabled and configured at 4MHz after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Power Manager (PM).

Some synchronous system clocks are active after Start-Up, allowing software execution. Synchronous system clocks that are running receive the 4MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

Related Links

13.8 PM – Power Manager

13.6.6.2.6 Peripheral Clock Masking

8.3.3 I/O Pins

After power-up, the I/O pins are tri-stated except PA30, which is pull-up enabled and configured as input.

Related Links

13.8 PM – Power Manager

8.3.4 Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x00000000. This points to the first executable address in the internal Flash memory. The code read from the internal Flash can be used to configure the clock system and clock sources. Refer to the ARM Architecture Reference Manual for more information on CPU startup (www.arm.com).

Related Links

13.8 PM – Power Manager

13.5 GCLK - Generic Clock Controller

13.10 OSC32KCTRL – 32KHz Oscillators Controller

8.4 Power-On Reset and Brown-Out Detector

The SAM R30 embeds three features to monitor, warn and/or reset the device:

- POR: Power-on Reset on VDDIN, VSWOUT and VDDIO
- BOD33: Brown-out detector on VSWOUT/VBAT
- Brown-out detector internal to the voltage regulator for VDDCORE. BOD12 is calibrated in production and its calibration parameters are stored in the NVM User Row. This data should not be changed if the User Row is written to in order to assure correct behavior.

8.4.1 Power-On Reset on VDDIN

VDDIN is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDDIN goes below the threshold voltage, the entire chip is reset.

8.4.2 Power-On Reset on VSWOUT

VSWOUT is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VSWOUT goes below the threshold voltage, the entire chip is reset.

8.4.3 Power-On Reset on VDDIO

VDDIO is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDDIO goes below the threshold voltage, all I/Os supplied by VSWOUT are reset.

8.4.4 Brown-Out Detector on VSWOUT/VBAT

BOD33 monitors VSWOUT or VBAT depending on configuration.

Related Links

13.11 SUPC – Supply Controller
13.11.6.3 Battery Backup Power Switch

8.4.5 Brown-Out Detector on VDDCORE

Once the device has started up, BOD12 monitors the internal VDDCORE.

Related Links

13.11 SUPC – Supply Controller
13.11.6.3 Battery Backup Power Switch

8.5 Performance Level Overview

By default, the device will start in Performance Level 0. This PL0 is aiming for the lowest power consumption by limiting logic speeds and the CPU frequency. As a consequence, all GCLK will have limited capabilities, and some peripherals and clock sources will not work or with limited capabilities:

List of peripherals/clock sources not available in PL0:

- USB (limited by logic frequency)
- DFLL48M

List of peripherals/clock sources with limited capabilities in PL0:

- All AHB/APB peripherals are limited by CPU frequency
- DPLL96M: may be able to generate 48MHz internally, but the output cannot be used by logic
- GCLK: the maximum frequency is by factor 4 compared to PL2
- SW interface: the maximum frequency is by factor 4 compared to PL2
- TC: the maximum frequency is by factor 4 compared to PL2
- TCC:the maximum frequency is by factor 4 compared to PL2
- SERCOM: the maximum frequency is by factor 4 compared to PL2

List of peripherals/clock sources with full capabilities in PL0:

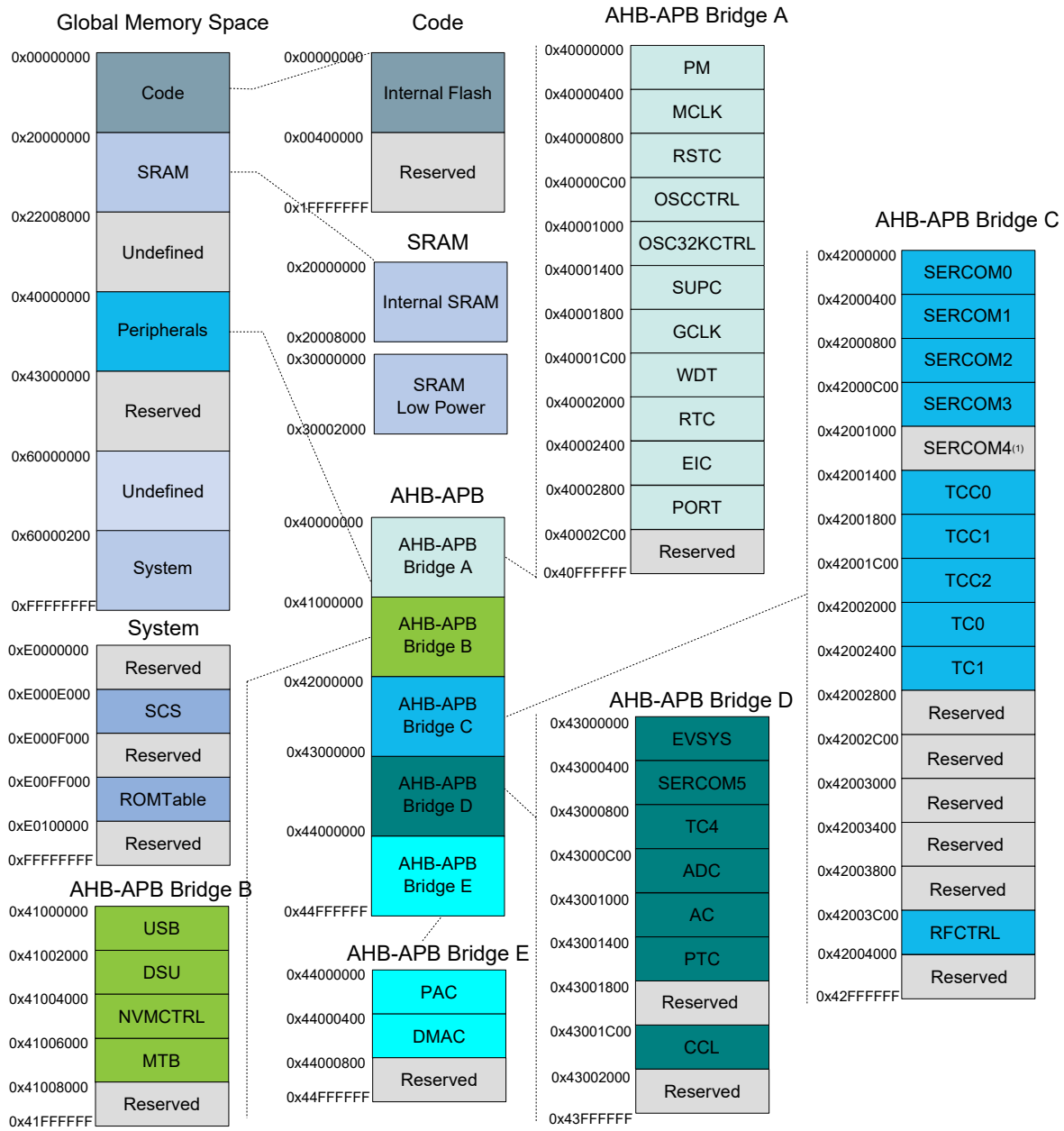
- AC
- ADC
- EIC
- OSC16M
- PTC
- All 32KHz clock sources and peripherals

Full functionality and capability will be ensured in PL2. When transitioning between performance levels, the Supply Controller (SUPC) will provide a configurable smooth voltage scaling transition.

Related Links

13.8 PM – Power Manager
13.11 SUPC – Supply Controller

9. Product Mapping



Note 1: SERCOM4 is internally connected to the AT86RF212B.

10. Memories

10.1 Embedded Memories

- Internal high-speed Flash with Read-While-Write (RWW) capability on a section of the array
- Internal high-speed RAM, single-cycle access at full speed
- Internal low-power RAM, single-cycle access at full speed

10.2 Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed, and they are never remapped in any way, even during boot. The 32-bit physical address space is mapped as follows:

Table 10-1. SAM R30 Physical Memory Map(1)

Memory	Start address	Size [KB]
		SAMR30x18
Embedded Flash	0x00000000	256
Embedded RWW section	0x00400000	8
Embedded SRAM	0x20000000	32
Embedded low-power SRAM	0x30000000	8
Peripheral Bridge A	0x40000000	64
Peripheral Bridge B	0x41000000	64
Peripheral Bridge C	0x42000000	64
Peripheral Bridge D	0x43000000	64
Peripheral Bridge E	0x44000000	64
IOBUS	0x60000000	0.5

1. 1. x = G or E.

Table 10-2. Flash Memory Parameters(1)

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAMR30x18	256	4096	64

1. 1. x = G or E.

Table 10-3. RWW Section Parameters(1)

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAMR30x18	8	128	64

1. 1. x = G or E.

10.3 NVM User Row Mapping

The Non Volatile Memory (NVM) User Row contains calibration data that are automatically read at device power-on.

The NVM User Row can be read at address 0x00804000.

To write the NVM User Row refer to the documentation of the NVMCTRL - Non-Volatile Memory Controller.

Note: When writing to the User Row, the new values do not get loaded by the other peripherals on the device until a device Reset occurs.

Table 10-4. NVM User Row Mapping

Bit Pos.	Name	Usage	Factory Setting	Related Peripheral Register
2:0	BOOTPROT	Used to select one of eight different bootloader sizes.	0x7	NVMCTRL
3	Reserved	—	0x1	—
6:4	EEPROM	Used to select one of eight different EEPROM sizes.	0x7	NVMCTRL
7	Reserved	—	0x1	—
13:8	BOD33 Level	BOD33 threshold level at power-on.	0x06	SUPC.BOD33
14	BOD33 Disable	BOD33 Disable at power-on.	0x0	SUPC.BOD33
16:15	BOD33 Action	BOD33 Action at power-on.	0x1	SUPC.BOD33
25:17	Reserved	Factory settings - do not change.	0x08F	-
26	WDT Enable	WDT Enable at power-on.	0x0	WDT.CTRLA
27	WDT Always-On	WDT Always-On at power-on.	0x0	WDT.CTRLA
31:28	WDT Period	WDT Period at power-on.	0xB	WDT.CONFIG
35:32	WDT Window	WDT Window mode time-out at power-on.	0xB	WDT.CONFIG
39:36	WDT EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on.	0xB	WDT.EWCTRL
40	WDT WEN	WDT Timer Window Mode Enable at power-on.	0x0	WDT.CTRLA
41	BOD33 Hysteresis	BOD33 Hysteresis configuration at power-on.	0x0	SUPC.BOD33
47:42	Reserved	Factory settings - do not change.	0x3E	—
63:48	LOCK	NVM Region Lock Bits.	0xFFFF	NVMCTRL

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller
13.11 SUPC – Supply Controller
13.12 WDT – Watchdog Timer

10.4 NVM Software Calibration Area Mapping

The NVM Software Calibration Area contains calibration data that are determined and written during production test. These calibration values should be read by the application software and written back to the corresponding register.

The NVM Software Calibration Area can be read at address 0x00806020.

The NVM Software Calibration Area can not be written.

Table 10-5. NVM Software Calibration Area Mapping

Bit Position	Name	Description
2:0	BIASREFBUF	ADC linearity. To be written to ADC CALIB.BIASREFBUF
5:3	BIASCOMP	ADC bias calibration. To be written to ADC CALIB.BIASCOMP
12:6	OSC32KCAL	OSC32K calibration. To be written to OSC32KCTRL OSC32K.CALIB
17:13	USB_TRANSN	USB pad calibration. To be written to USB PADCAL.TRANSN
22:18	USB_TRANSP	USB pad calibration. To be written to USB PADCAL.TRANSP
25:23	USB_TRIM	USB pad calibration. To be written to USB PADCAL.TRIM
31:26	DFLL48M_COARSE_CAL	DFLL48M coarse calibration. To be written to OSCCTRL DFLLVAL.COARSE

Related Links

13.27.8.22 CALIB
13.25.8.1.6 PADCAL
13.9.8.8 DFLLVAL

10.5 NVM Temperature Log Row

The NVM Temperature Log Row contains calibration data that are determined and written during production test. These calibration values are required for calculating the temperature from measuring the temperature sensor in the Supply Controller (SUPC) by the ADC.

The NVM Temperature Log Row can be read at address 0x00806030.

The NVM Temperature Log Row cannot be written.

Table 10-6. Temperature Log Row Content

Bit Position	Name	Description
7:0	ROOM_TEMP_VAL_INT	Integer part of room temperature in °C
11:8	ROOM_TEMP_VAL_DEC	Decimal part of room temperature
19:12	HOT_TEMP_VAL_INT	Integer part of hot temperature in °C
23:20	HOT_TEMP_VAL_DEC	Decimal part of hot temperature
31:24	ROOM_INT1V_VAL	2's complement of the internal 1V reference drift at room temperature (versus a 1.0-centered value)
39:32	HOT_INT1V_VAL	2's complement of the internal 1V reference drift at hot temperature (versus a 1.0-centered value)
51:40	ROOM_ADC_VAL	Temperature sensor 12-bit ADC conversion at room temperature
63:52	HOT_ADC_VAL	Temperature sensor 12-bit ADC conversion at hot temperature

References:

13.27.6.3.2 Device Temperature Measurement

15.10.8 Temperature Sensor Characteristics

10.6 Serial Number

Each device has a unique 128-bit serial number which is a concatenation of four 32-bit words contained at the following addresses:

Word 0: 0x0080A00C

Word 1: 0x0080A040

Word 2: 0x0080A044

Word 3: 0x0080A048

The uniqueness of the serial number is guaranteed only when using all 128 bits.

11. Processor and Architecture

11.1 Cortex M0+ Processor

The SAM R30 implements the ARM®Cortex™-M0+ processor, based on the ARMv6 Architecture and Thumb®-2 ISA. The Cortex M0+ is 100% instruction set compatible with its predecessor, the Cortex-M0 core, and upward compatible to Cortex-M3 and M4 cores. The implemented ARM Cortex-M0+ is revision r0p1. For more information refer to www.arm.com

11.1.1 Cortex M0+ Configuration

Table 11-1. Cortex M0+ Configuration in SAM R30

Features	Cortex M0+ options	SAM R30 configuration
Interrupts	External interrupts 0-32	29
Data endianness	Little-endian or big-endian	Little-endian
SysTick timer	Present or absent	Present
Number of watchpoint comparators	0, 1, 2	2
Number of breakpoint comparators	0, 1, 2, 3, 4	4
Halting debug support	Present or absent	Present
Multiplier	Fast or small	Fast (single cycle)
Single-cycle I/O port	Present or absent	Present
Wake-up interrupt controller	Supported or not supported	Not supported
Vector Table Offset Register	Present or absent	Present
Unprivileged/Privileged support	Present or absent	Absent - All software run in privileged mode only
Memory Protection Unit	Not present or 8-region	Not present
Reset all registers	Present or absent	Absent
Instruction fetch width	16-bit only or mostly 32-bit	32-bit

The ARM Cortex-M0+ core has two bus interfaces:

- Single 32-bit AMBA-3 AHB-Lite system interface that provides connections to peripherals and all system memory including Flash memory and RAM
- Single 32-bit I/O port bus interfacing to the PORT with 1-cycle loads and stores

11.1.1.1 Cortex M0+ Peripherals

- System Control Space (SCS)
  - The processor provides debug through registers in the SCS. Refer to the Cortex-M0+ Technical Reference Manual for details (www.arm.com)
- Nested Vectored Interrupt Controller (NVIC)
  - External interrupt signals connect to the NVIC, and the NVIC prioritizes the interrupts. Software can set the priority of each interrupt. The NVIC and the Cortex-M0+ processor core are closely coupled, providing low latency interrupt processing and efficient processing of late arriving interrupts. Refer to the Cortex-M0+ Technical Reference Manual for details (www.arm.com).

Note: When the CPU frequency is much higher than the APB frequency it is recommended to insert a memory read barrier after each CPU write to registers mapped on the APB. Failing to do so in such conditions may lead to unexpected behavior such as re-entering a peripheral interrupt handler just after leaving it.

- System Timer (SysTick)
  - The System Timer is a 24-bit timer clocked by CLK_CPU that extends the functionality of both the processor and the NVIC. Refer to the Cortex-M0+ Technical Reference Manual for details (www.arm.com).
- System Control Block (SCB)
  - The System Control Block provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions. Refer to the Cortex-M0+ Devices Generic User Guide for details (www.arm.com)
- Micro Trace Buffer (MTB)
  - The CoreSight MTB-M0+ (MTB) provides a simple execution trace capability to the Cortex-M0+ processor. Refer to section MTB-Micro Trace Buffer and the CoreSight MTB-M0+ Technical Reference Manual for details (www.arm.com).

Related Links

11.2 Nested Vector Interrupt Controller

11.1.1.2 Cortex M0+ Address Map

Table 11-2. Cortex-M0+ Address Map

Address	Peripheral
0xE000E000	System Control Space (SCS)
0xE000E010	System Timer (SysTick)
0xE000E100	Nested Vectored Interrupt Controller (NVIC)
0xE000ED00	System Control Block (SCB)
0x41006000	Micro Trace Buffer (MTB)

11.1.1.3 I/O Interface

The device allows direct access to PORT registers. Accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, so the Cortex M0+ processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O access to be sustained for as long as necessary.

Related Links

13.17 PORT - I/O Pin Controller

13.17.5.10 CPU Local Bus

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM R30 supports 32 interrupt lines with four different priority levels. For more details, refer to the Cortex-M0+ Technical Reference Manual (www.arm.com).

11.2.2 Interrupt Line Mapping

Each of the 23 interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a 1 to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing 1 to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Table 11-3. Interrupt Line Mapping

Peripheral source	NVIC line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager MCLK - Main Clock OSCCTRL - Oscillators Controller OSC32KCTRL - 32KHz Oscillators Controller SUPC - Supply Controller PAC - Protecion Access Controller	0
WDT – Watchdog Timer	1
RTC – Real Time Counter	2
EIC – External Interrupt Controller	3
NVMCTRL – Non-Volatile Memory Controller	4
DMAC - Direct Memory Access Controller	5
USB - Universal Serial Bus	6
EVSYS – Event System	7
SERCOM0 – Serial Communication Interface 0	8
SERCOM1 – Serial Communication Interface 1	9
SERCOM2 – Serial Communication Interface 2	10
SERCOM3 – Serial Communication Interface 3	11
SERCOM4 – Serial Communication Interface 4	12
SERCOM5 – Serial Communication Interface 5	13
TCC0 – Timer Counter for Control 0	14
TCC1 – Timer Counter for Control 1	15
TCC2 – Timer Counter for Control 2	16
TC0 – Timer Counter 0	17
TC1 – Timer Counter 1	18
TC4 – Timer Counter 4	19
ADC – Analog-to-Digital Converter	20
AC – Analog Comparator	21
PTC – Peripheral Touch Controller	22

11.3 Micro Trace Buffer

11.3.1 Features

- Program flow tracing for the Cortex-M0+ processor
- MTB SRAM can be used for both trace and general purpose storage by the processor
- The position and size of the trace buffer in SRAM is configurable by software
- CoreSight compliant

11.3.2 Overview

When enabled, the MTB records the changes in program flow that are reported by the Cortex-M0+ processor over the execution trace interface. This interface is shared between the Cortex-M0+ processor and the CoreSight MTB-M0+. The information is stored by the MTB in the SRAM as trace packets. An off-chip debugger can extract the trace information using the Debug Access Port to read the trace information from the SRAM. The debugger can then reconstruct the program flow from this information.

The MTB stores trace information into the SRAM and gives the processor access to the SRAM simultaneously. The MTB ensures that trace write accesses have priority over processor accesses.

An execution trace packet consists of a pair of 32-bit words that the MTB generates when it detects a non-sequential change of the program pounter (PC) value. A non-sequential PC change can occur during branch instructions or during exception entry. See the CoreSight MTB-M0+ Technical Reference Manual for more details on the MTB execution trace packet format.

Tracing is enabled when the MASTER.EN bit in the Master Trace Control Register is 1. There are various ways to set the bit to 1 to start tracing, or to 0 to stop tracing. See the CoreSight Cortex-M0+ Technical Reference Manual for more details on the Trace start and stop and for a detailed description of the MTB's MASTER register. The MTB can be programmed to stop tracing automatically when the memory fills to a specified watermark level or to start or stop tracing by writing directly to the MASTER.EN bit. If the watermark mechanism is not being used and the trace buffer overflows, then the buffer wraps around overwriting previous trace packets.

The base address of the MTB registers is 0x41006000; this address is also written in the CoreSight ROM Table. The offset of each register from the base address is fixed and as defined by the CoreSight MTB-M0+ Technical Reference Manual. The MTB has four programmable registers to control the behavior of the trace features:

- POSITION: Contains the trace write pointer and the wrap bit
- MASTER: Contains the main trace enable bit and other trace control fields
- FLOW: Contains the WATERMARK address and the AUTOSTOP and AUTOHALT control bits
- BASE: Indicates where the SRAM is located in the processor memory map. This register is provided to enable auto discovery of the MTB SRAM location by a debug agent

See the CoreSight MTB-M0+ Technical Reference Manual for a detailed description of these registers.

11.4 High-Speed Bus System

11.4.1 Features

High-Speed Bus Matrix has the following features:

- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different masters to different slaves
- 32-bit data bus
- Operation at a one-to-one clock frequency with the bus masters

H2LBRIDGE has the following features:

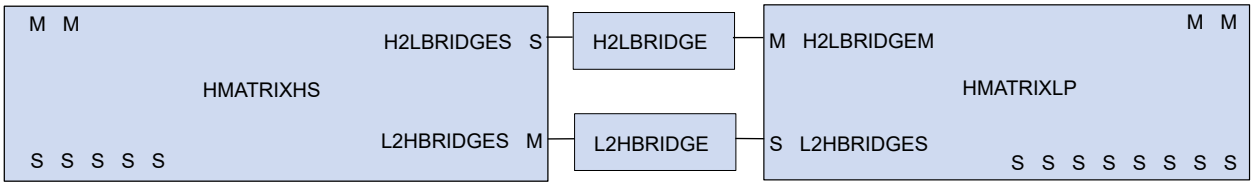
- LP clock division support
- Write: Posted-write FIFO of 3 words, no bus stall until it is full
- Write: 1 cycle bus stall when full when LP clock is not divided

- 2 stall cycles on read when LP clock is not divided
- Ultra low latency mode:
  - Suitable when the HS clock frequency is not above half the maximum device clock frequency
  - Removes all intrinsic bridge stall cycles (except those needed for LP clock ratio adaptation)
  - Enabled by writing a '1' in 0x41008120 using a 32-bit write access

L2HBRIDGE has the following features:

- LP clock division support
- Write: Posted-write FIFO of 1 word, no bus stall until it is full
- Write: 1 cycle bus stall when full when LP clock is not divided
- 2 stall cycles on read when LP clock is not divided
- ultra low latency mode:
  - Suitable when the HS clock frequency is not above half the maximum device clock frequency
  - Removes all intrinsic bridge stall cycles (except those needed for LP clock ratio adaptation)
  - Enabled by writing a '1' in 0x41008120 using a 32-bit write access

Figure 11-1. High-Speed Bus System Components



11.4.2 Configuration

Figure 11-2. Master-Slave Relations High-Speed Bus Matrix

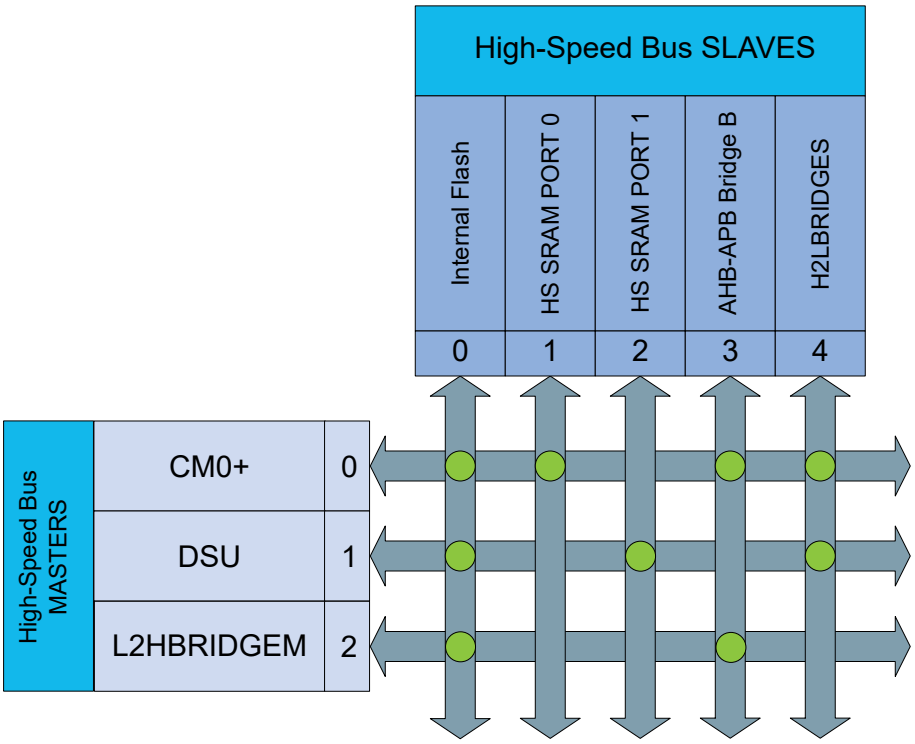


Figure 11-3. Master-Slave Relations Low-Power Bus Matrix

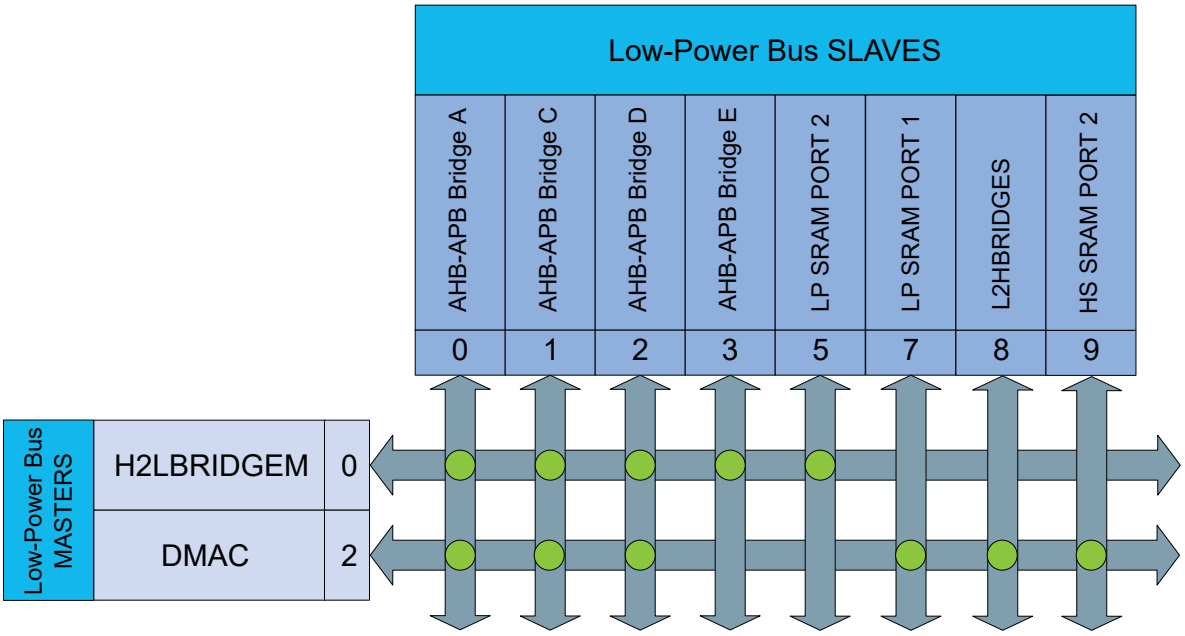


Table 11-4. High-Speed Bus Matrix Masters

High-Speed Bus Matrix Masters	Master ID
CM0+ - Cortex M0+ Processor	0
DSU - Device Service Unit	1
L2HBRIDGEM - Low-Power to High-Speed bus matrix AHB to AHB bridge	2

Table 11-5. High-Speed Bus Matrix Slaves

High-Speed Bus Matrix Slaves	Slave ID
Internal Flash Memory	0
HS SRAM Port 0 - CM0+ Access	1
HS SRAM Port 1 - DSU Access	2
AHB-APB Bridge B	3
H2LBRIDGES - High-Speed to Low-Power bus matrix AHB to AHB bridge	4

Table 11-6. Low-Power Bus Matrix Masters

Low-Power Bus Matrix Masters	Master ID
H2LBRIDGEM - High-Speed to Low-Power bus matrix AHB to AHB bridge	0
DMAC - Direct Memory Access Controller - Data Access	2

Table 11-7. Low-Power Bus Matrix Slaves

Low-Power Bus Matrix Slaves	Slave ID
AHB-APB Bridge A	0

.....continued	
Low-Power Bus Matrix Slaves	**Slave ID**
AHB-APB Bridge C	1
AHB-APB Bridge D	2
AHB-APB Bridge E	3
LP SRAM Port 2- H2LBRIDGEM access	5
LP SRAM Port 1- DMAC access	7
L2HBRIDGES - Low-Power to High-Speed bus matrix AHB to AHB bridge	8
HS SRAM Port 2- HMATRIXLP access	9

11.4.3 SRAM Quality of Service

To ensure that masters with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the masters for different types of access.

The Quality of Service (QoS) level is independently selected for each master accessing the RAM. For any access to the RAM, the RAM also receives the QoS level. The QoS levels and their corresponding bit values for the QoS level configuration are shown in the following table.

Table 11-8. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

If a master is configured with QoS level DISABLE (0x0) or LOW (0x1) there will be a minimum latency of one cycle for the RAM access.

The priority order for concurrent accesses are decided by two factors. First, the QoS level for the master and second, a static priority given by the port ID. The lowest port ID has the highest static priority. See the tables below for details.

The MTB has a fixed QoS level HIGH (0x3).

The CPU QoS level can be written/read, using 32-bit access only, at address 0x41008114 bits [1:0]. Its reset value is 0x3.

Refer to different master QOSCTRL registers for configuring QoS for the other masters (USB, DMAC).

Table 11-9. HS SRAM Port Connections QoS

HS SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
MTB - Micro Trace Buffer	4	Direct	STATIC-3	0x3
USB - Universal Serial Bus	3	Direct	IP-QOSCTRL	0x3
HMATRIXLP - Low-Power Bus Matrix	2	Bus Matrix	0x44000934(1), bits[1:0]	0x2
DSU - Device Service Unit	1	Bus Matrix	0x4100201C(1)	0x2
CM0+ - Cortex M0+ Processor	0	Bus Matrix	0x41008114(1), bits[1:0]	0x3

Note:

1. Using 32-bit access only.

Table 11-10. LP SRAM Port Connections QoS

LP SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
DMAC - Direct Memory Access Controller - Write-Back Access	5, 6	Direct	IP-QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Fetch Access	3, 4	Direct	IP-QOSCTRL.FQOS	0x2
H2LBRIDGEM - HS to LP bus matrix AHB to AHB bridge	2	Bus Matrix	0x44000924(1), bits[1:0]	0x2
DMAC - Direct Memory Access Controller - Data Access	1	Bus Matrix	IP-QOSCTRL.DQOS	0x2

Note:

1. Using 32-bit access only.

12. Application Schematic Introduction

The SAM R30 Application schematic has to provide the environment for both integrated circuits inside the package. The micro controller as well as the radio have integrated LDO's to provide the required core voltages. To achieve the full radio performance the application layout has to take the noise decoupling in between the analog radio part and the digital processor and signal processing power domains into account. To avoid noise interference with the radio part, the switch regulator is not available in this version of SAM L21.

12.1 SAM R30 Basic Application Schematic

Mandatory circuit elements to operate the SAM R30 system in package

The schematic drawing shows the mandatory external circuit for the SAM R30 system. This circuit is required in the same way for the package configurations with 32 or 48 pins. The package variants are different in terms of the available controller IO pins. These pins are not shown in the application schematic drawing.

Unused pins that are not shown in the drawing can be left open. Keep in mind that the software has to initialize floating pins to a valid static state. For an open IO this can be a configuration as output with ether high or low level output. For a floating input pins it is recommended to activate internal pull resistors to ensure a steady high or low state. Input pins that are floating at levels above the maximum low voltage or minimum high voltage will cause additional supply current consumption. It will not be possible to achieve the specified sleep current values with floating input pins within the system. This is also valid for input pins that are wired to an unused connector. Consider to add large pull resistors (e.g. 1MΩ) to input signals where there is no guaranteed driving source.

The paddle (AKA heatsink) on the bottom of the QFN package is an active ground pin. On SAM R30 the paddle is bonded to the digital ground substrate of the CPU, or GND signal. Because the SAM R30 is a low power device the thermal output is minimal. However, a solid ground connection to the paddle is necessary. The paddle land area should be connected to the inner digital ground planes with several vias. The GND pins can be directly connected to the paddle land area as long as at least one via to the system digital ground is provided. The GNDANA signal should be treated as an independent ground domain. The GNDANA pins adjacent the RFP and RFN ports need to have a direct low-impedance connection to the RF ground plane of the system. This increases both the TX and RX signal strength and reduces phase noise. PCB layout should be separate GND and GNDANA structures to reduce common impedance coupling of ground return currents. GND and GNDANA should only be connected at one single point in a ‘star ground’ pattern. The connection point can be near the IC or Power Supply.

Please consider additional information in the pin description and signal description sections during schematic design.

Figure 12-1. SAM R30 Basic Application Schematic

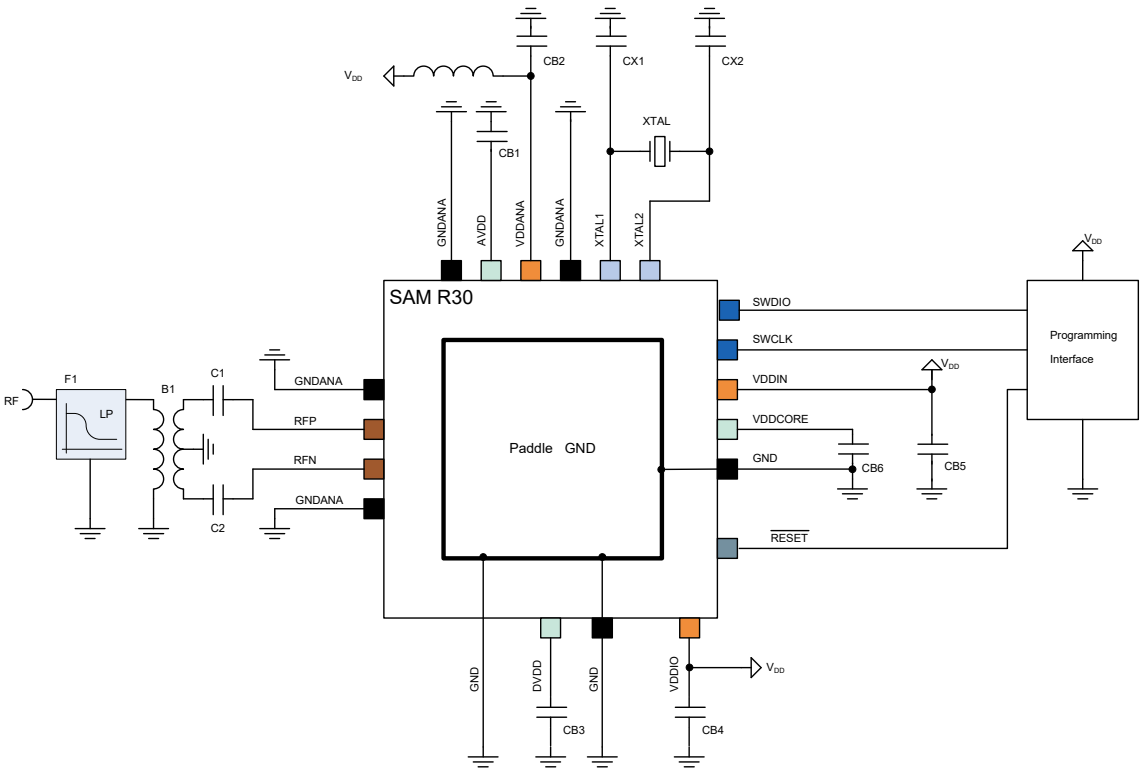


Table 12-1. Example Bill of Materials (BoM) for the Basic Application Schematic.

Symbol	Description	Value	Manufacturer	Part Number	Comment
B1	SMD balun	800 – 1000MHz	Wuerth JTI	748431090 0900BL18B100	Differential to singe ended transformer. Does not contain any harmonic filter.
F1	Low Pass Filter	Specific for application frequency range			1)
B1, F1, C1, C2	Balun/Filter combination	863 – 928MHz	JTI	0896BM15A0032	Custom matching for AT86RF212B
		779 – 787MHz	JTI	0783FB15A0100	validation pending
C1, C2	C0G, 5%, 0402, 50V	100pF	Murata	GRM1555C1H101JA01	DC blocking 2)
CX1, CX2	C0G, 5%, 0402, 50V	10pF	Murata	GRM1555C1H100JA01	Crystal load capacitor for XTAL with CL=9pF
CB1	X5R, 20%, 10V, 0402	1uF	Murata	GRM153R61A105ME95	Transceiver analog core voltage bypass

.....continued					
Symbol	**Description**	**Value**	**Manufacturer**	**Part Number**	**Comment**
CB3	X5R, 20%, 10V, 0402	1uF	Murata	GRM153R61A105ME95	Transceiver digital core voltage bypass
CB6	X5R, 20%, 10V, 0402	1uF	Murata	GRM153R61A105ME95	CPU digital core voltage bypass
CB2	X5R, 20%, 10V, 0402	1uF	Murata	GRM153R61A105ME95	Analog supply voltage input bypass
CB4, CB5	X5R, 20%, 6.3V, 0402	4.7uF	Murata	GRM155R60J475ME87	Digital supply voltage input bypass
XTAL	Crystal, 16MHz, 10ppm@25°C, 15ppm over temperature, CL=9pF	CX-4025 16MHz SX-4025 16MHz SMD 2x2.5mm 16MHz	ACAL Taitjen Siward TAI-SAW Technology	XWBBPL-F-1 A207-011 TZ1670C	3)

Notes: 1)
The required filter depends on the deployment region. For Sub-1GHz countries have defined individual limits for spurious emissions and different frequency ranges for license free radio applications.

- Chinese WPAN band from 779 to 787MHz
- European SRD band from 863 to 870MHz
- North American ISM band from 902 to 928MHz
- Japanese band from 915 to 930MHz

Note: 2)
For technical reasons the internal bias voltage is available at the RF pins RFP and RFN. This voltage shall not be loaded with a considerable current. It the chosen balun has a DC path to ground or the single ended output then C1 and C2 are required to isolate potential DC current.

Note: 3)

The crystal start-up time will strongly depend on the ESR parameter. The smaller the crystal, the higher the typical ESR value. A small crystal with eg. 80Ohm ESR may take up to 2ms for stable operation while a larger part with an ESR of 30Ohm may start faster than 1ms. This may be considered when selecting crystals for low power applications where the start-up time matters for the over all power consumption. A larger CL value can increase the immunity against EMI or xtalk. At the same time the start-up time is increased. In case of a changed CL value, the CX capacitors need to be changed accordingly.

12.2 Extended Feature Set Application Schematic

Application circuit for extended RF front end functions.

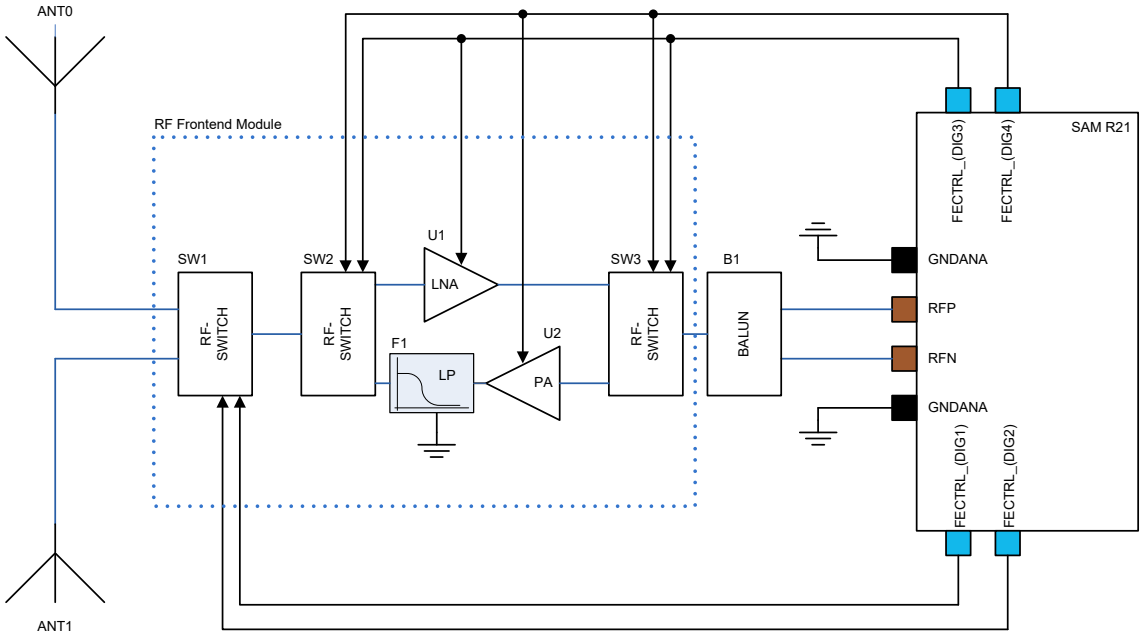
SAM R30 has build in hardware support for antenna diversity and active range extension front end circuits. The digital control signal outputs from RF212B are multiplexed to SAM L21 port pins. The Front End Control Signal Interface allows the free selection of six alternative IO pins for the four RF212B DIG signals. For that reason no particular pin number had been assigned in the schematic drawing. The internal multiplexer allows a pin selection for best application match instead.

A large number of higher integrated front end modules are available in the market for many different application scenarios. The schematic drawing below shows a contrived front end that would make use of all four DIG signals. DIG4 is an inverted signal of DIG3 and in the same way DIG2 is an inverted signal of DIG1. For DIG2 there is also a function as a time stamping trigger or interrupt available.

With the signals DIG1 and DIG2 provision is made for antenna diversity operation. The detailed description for the antenna diversity function and its configuration can be found in the extended feature set section for the RF212B. RF switch ICs do often require a signal pair with one line inverted. Such switches can be directly controlled without additional glue logic. The switch on/off time should be selected to be below 100ns.

The signals DIG3 and DIG4 are designated to switch the front end functions in between receive and transmit. Typical functions are path selection as well as PA or LNA enable and disable functions.

Figure 12-2. SAM R30 RF front end control signal schematic drawing



13. Reference Guide - SAM L21

The SAM R30 system in package contains a SAM L21 die. It is recommended to use available documentation, software sources and application notes for the SAM L21 as additional reference information.

13.1 PAC - Peripheral Access Controller

13.1.1 Overview

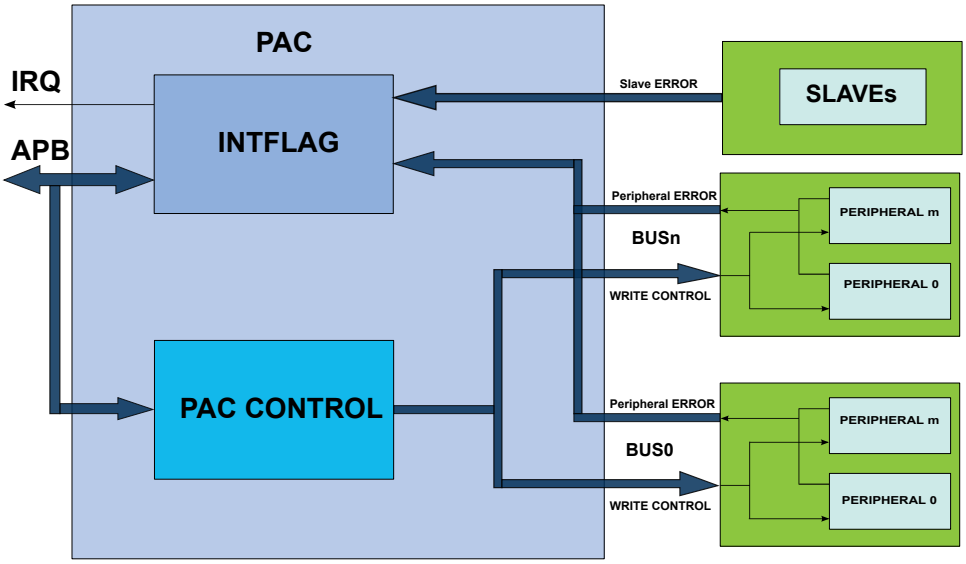
The Peripheral Access Controller provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the slave bus level, when an access to a non-existing address is detected.

13.1.2 Features

- Manages write protection access and reports access errors for the peripheral modules or bridges

13.1.3 Block Diagram

Figure 13-1. PAC Block Diagram



13.1.4 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.1.4.1 IO Lines

Not applicable.

13.1.4.2 Power Management

The PAC can continue to operate in any sleep mode where the selected source clock is running. The PAC interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.1.4.3 Clocks

The PAC bus clock (CLK_PAC_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_PAC_APB can be found in the related links.

Related Links

13.6 MCLK – Main Clock

13.6.6.2.6 Peripheral Clock Masking

13.1.4.4 DMA

Not applicable.

13.1.4.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the PAC interrupt requires the Interrupt Controller to be configured first.

Table 13-1. Interrupt Lines

Instances	NVIC Line
PAC	PACERR

Related Links

11.2 Nested Vector Interrupt Controller

13.1.4.6 Events

The events are connected to the Event System, which may need configuration.

Related Links

13.18 EVSYS – Event System

13.1.4.7 Debug Operation

When the CPU is halted in debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

13.1.4.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Write Control (WRCTRL) register
- AHB Slave Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.1.5 Functional Description

13.1.5.1 Principle of Operation

The Peripheral Access Control module allows the user to set a write protection on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral's protection can be set, cleared or locked at the user discretion. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, slaves bus errors can be also reported in the cases where reserved area is accessed by the application.

13.1.5.2 Basic Operation

13.1.5.2.1 Initialization, Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

13.1.5.2.2 Operations

The PAC module allows the user to set, clear or lock the write protection status of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. Refer to the 13.1.5.2.3 Peripheral Access Errors for details.

The PAC module also report the errors occurring at slave bus level when an access to reserved area is detected. AHB Slave Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding slave. Refer to the 13.1.5.2.6 AHB Slave Bus Errors for details.

13.1.5.2.3 Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- Protected write: To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as “PAC Write-Protection” in the module’s datasheet can be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

Related Links

13.4.3 Register Synchronization

13.1.5.2.4 Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields; WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be “clear protection”, “set protection” and “set and lock protection bit”.

The “clear protection” operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

13.1.5.2.5 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation.

In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection. This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when

the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGn.PAC flag.

13.1.5.2.6 AHB Slave Bus Errors

The PAC module reports errors occurring at the AHB Slave bus level. These errors are generated when an access is performed at an address where no slave (bridge or peripheral) is mapped . These errors are reported in the corresponding bits of the INTFLAGAHB register.

13.1.5.2.7 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

13.1.5.3 DMA Operation

Not applicable.

13.1.5.4 Interrupts

The PAC has the following interrupt source:

- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the PAC is reset. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

11.2 Nested Vector Interrupt Controller

13.1.5.5 Events

The PAC can generate the following output event:

- Error (ERR): Generated when one of the interrupt flag registers bits is set

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

13.1.5.6 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus master (CPU, DMA) is running. The PAC will continue to catch access errors from the module and generate interrupts or events.

13.1.5.7 Synchronization

Not applicable.

13.1.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	WRCTRL	7:0	PERID[7:0]							
		15:8	PERID[15:8]							
		23:16	KEY[7:0]							
		31:24								
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	7:0				H2LBRIDGES	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
		15:8								
		23:16	LPRAMDMAC		LPRAMHS		HPB4	HPB3	HPB2	HPB0
		31:24							HSRAMLP	L2HBRIDGES
0x14	INTFLAGA	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
		15:8				DSUSTANDBY		PORT	EIC	RTC
		23:16								
		31:24								
0x18	INTFLAGB	7:0				HMATRIXHS	MTB	NVMCTRL	DSU	USB
		15:8								
		23:16								
		31:24								
0x1C	INTFLAGC	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
		15:8							TC1	TC0
		23:16								
		31:24								
0x20	INTFLAGD	7:0	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
		15:8								
		23:16								
		31:24								
0x24	INTFLAGE	7:0						HMATRIXLP	DMAC	PAC
		15:8								
		23:16								
		31:24								
0x28 ... 0x33	Reserved									
0x34	STATUSA	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
		15:8				DSUSTANDBY		PORT	EIC	RTC
		23:16								
		31:24								
0x38	STATUSB	7:0				HMATRIXHS	MTB	NVMCTRL	DSU	USB
		15:8								
		23:16								
		31:24								
0x3C	STATUSC	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
		15:8							TC1	TC0
		23:16								
		31:24								

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	STATUSD	7:0	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
		15:8								
		23:16								
		31:24								
0x44	STATUSE	7:0						HMATRIXLP	DMAC	PAC
		15:8								
		23:16								
		31:24								

13.1.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the related links.

Related Links

13.4.3 Register Synchronization

13.1.7.1 Write Control

Name: WRCTRL
Offset: 0x0
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0] Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock the peripheral write control until the next hardware reset

Bits 15:0 – PERID[15:0] Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is calculated following formula:
$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$

Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number:

Table 13-2. PERID Values

Periph. Bridge Name	BridgeNumber	PERID Values
A	0	0+N
B	1	32+N
C	2	64+N
D	3	96+N
E	4	128+N

13.1.7.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								ERREO
Access								R/W
Reset								0

Bit 0 – ERREO Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, an event will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Value	Description
0	Peripheral Access Error Event Output is disabled.
1	Peripheral Access Error Event Output is enabled.

13.1.7.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Disable
This bit indicates that the Peripheral Access Error Interrupt is disabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

13.1.7.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR Peripheral Access Error Interrupt Enable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

13.1.7.5 AHB Slave Bus Interrupt Flag Status and Clear

Name: INTFLAGAHB
Offset: 0x10
Reset: 0x000000
Property: –

This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit	31	30	29	28	27	26	25	24
							HSRAMLP	L2HBRIDGES
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	LPRAMDMAC		LPRAMHS		HPB4	HPB3	HPB2	HPB0
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				H2LBRIDGES	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 25 – HSRAMLP Interrupt Flag for SLAVE HSRAMLP
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 24 – L2HBRIDGES Interrupt Flag for SLAVE L2HBRIDGES
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 23 – LPRAMDMAC Interrupt Flag for SLAVE LPRAMDMAC
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 21 – LPRAMHS Interrupt Flag for SLAVE LPRAMHS
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 19 – HPB4 Interrupt Flag for SLAVE HPB4
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 18 – HPB3 Interrupt Flag for SLAVE HPB3
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 17 – HPB2 Interrupt Flag for SLAVE HPB2
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 16 – HPB0 Interrupt Flag for SLAVE HPB0
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 4 – H2LBRIDGES Interrupt Flag for SLAVE H2LBRIDGES
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 3 – HPB1 Interrupt Flag for SLAVE HPB1
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 2 – HSRAMDSU Interrupt Flag for SLAVE HSRAMDSU
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 1 – HSRAMCM0P Interrupt Flag for SLAVE HSRAMCM0P
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Bit 0 – FLASH Interrupt Flag for SLAVE FLASH
This flag is cleared by writing a '1' to the flag. This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

13.1.7.6 Peripheral Interrupt Flag Status and Clear A

Name: INTFLAGA
Offset: 0x14
Reset: 0x000000
Property: –

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGA interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DSUSTANDBY		PORT	EIC	RTC
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – DSUSTANDBY Interrupt Flag for DSUSTANDBY

Bit 10 – PORT Interrupt Flag for PORT

Bit 9 – EIC Interrupt Flag for EIC

Bit 8 – RTC Interrupt Flag for RTC

Bit 7 – WDT Interrupt Flag for WDT

Bit 6 – GCLK Interrupt Flag for GCLK

Bit 5 – SUPC Interrupt Flag for SUPC

Bit 4 – OSC32KCTRL Interrupt Flag for OSC32KCTRL

Bit 3 – OSCCTRL Interrupt Flag for OSCCTRL

Bit 2 – RSTC Interrupt Flag for RSTC

Bit 1 – MCLK Interrupt Flag for MCLK

Bit 0 – PM Interrupt Flag for PM

13.1.7.7 Peripheral Interrupt Flag Status and Clear B

Name: INTFLAGB
Offset: 0x18
Reset: 0x000000
Property: –

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGB interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS	MTB	NVMCTRL	DSU	USB
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – HMATRIXHS Interrupt Flag for HMATRIXHS

Bit 3 – MTB Interrupt Flag for MTB

Bit 2 – NVMCTRL Interrupt Flag for NVMCTRL

Bit 1 – DSU Interrupt Flag for DSU

Bit 0 – USB Interrupt Flag for USB

13.1.7.8 Peripheral Interrupt Flag Status and Clear C

Name: INTFLAGC
Offset: 0x1C
Reset: 0x000000
Property: —

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGC interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TC1	TC0
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8, 9 – TC Interrupt Flag for TCn [n = 1..0]

Bits 5, 6, 7 – TCC Interrupt Flag for TCCn [n = 2..0]

Bits 0, 1, 2, 3, 4 – SERCOM Interrupt Flag for SERCOMn [n = 4..0]

13.1.7.9 Peripheral Interrupt Flag Status and Clear D

Name: INTFLAGD
Offset: 0x20
Reset: 0x000000
Property: –

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGD bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGD interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – CCL Interrupt Flag for CCL

Bit 5 – PTC Interrupt Flag for PTC

Bit 4 – AC Interrupt Flag for AC

Bit 3 – ADC Interrupt Flag for ADC

Bit 2 – TC4 Interrupt Flag for TC4

Bit 1 – SERCOM5 Interrupt Flag for SERCOM5

Bit 0 – EVSYS Interrupt Flag for EVSYS

13.1.7.10 Peripheral Interrupt Flag Status and Clear E

Name: INTFLAGE
Offset: 0x24
Reset: 0x000000
Property: –

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGE bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGE interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						HMATRIXLP	DMAC	PAC
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – HMATRIXLP Interrupt Flag for HMATRIXLP

Bit 1 – DMAC Interrupt Flag for DMAC

Bit 0 – PAC Interrupt Flag for PAC

13.1.7.11 Peripheral Write Protection Status A

Name: STATUSA
Offset: 0x34
Reset: 0x000000
Property: –

Writing to this register has no effect.

Reading STATUS register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DSUSTANDBY		PORT	EIC	RTC
Access				R		R	R	R
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 12 – DSUSTANDBY Peripheral DSUSTANDBY Write Protection Status

Bit 10 – PORT Peripheral PORT Write Protection Status

Bit 9 – EIC Peripheral EIC Write Protection Status

Bit 8 – RTC Peripheral RTC Write Protection Status

Bit 7 – WDT Peripheral WDT Write Protection Status

Bit 6 – GCLK Peripheral GCLK Write Protection Status

Bit 5 – SUPC Peripheral SUPC Write Protection Status

Bit 4 – OSC32KCTRL Peripheral OSC32KCTRL Write Protection Status

Bit 3 – OSCCTRL Peripheral OSCCTRL Write Protection Status

Bit 2 – RSTC Peripheral RSTC Write Protection Status

Bit 1 – MCLK Peripheral MCLK Write Protection Status

Bit 0 – PM Peripheral ATW Write Protection Status

13.1.7.12 Peripheral Write Protection Status B

Name: STATUSB
Offset: 0x38
Reset: 0x000000
Property: –

Writing to this register has no effect.

Reading STATUS register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS	MTB	NVMCTRL	DSU	USB
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Write Protection Status

Bit 3 – MTB Peripheral MTB Write Protection Status

Bit 2 – NVMCTRL Peripheral NVMCTRLWrite Protection Status

Bit 1 – DSU Peripheral DSU Write Protection Status

Bit 0 – USB Peripheral USB Write Protection Status

13.1.7.13 Peripheral Write Protection Status C

Name: STATUSC
Offset: 0x3C
Reset: 0x00000000
Property: —

Writing to this register has no effect.

Reading STATUS register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TC1	TC0
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 8, 9 – TCn Peripheral TCn Write Protection Status [n = 1..0]

Bits 5, 6, 7 – TCCn Peripheral TCC Write Protection Status

Bits 0, 1, 2, 3, 4 – SERCOMn Peripheral SERCOMn Write Protection Status [n = 4..0]

13.1.7.14 Peripheral Write Protection Status D

Name: STATUSD
Offset: 0x40
Reset: 0x000000
Property: –

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit 7 – CCL Peripheral CCL Write Protection Status

Bit 5 – PTC Peripheral PTC Write Protection Status

Bit 4 – AC Peripheral AC Write Protection Status

Bit 3 – ADC Peripheral ADC Write Protection Status

Bit 2 – TC4 Peripheral TC4 Write Protection Status

Bit 1 – SERCOM5 Peripheral SERCOM5 Write Protection Status

Bit 0 – EVSYS Peripheral EVSYS Write Protection Status

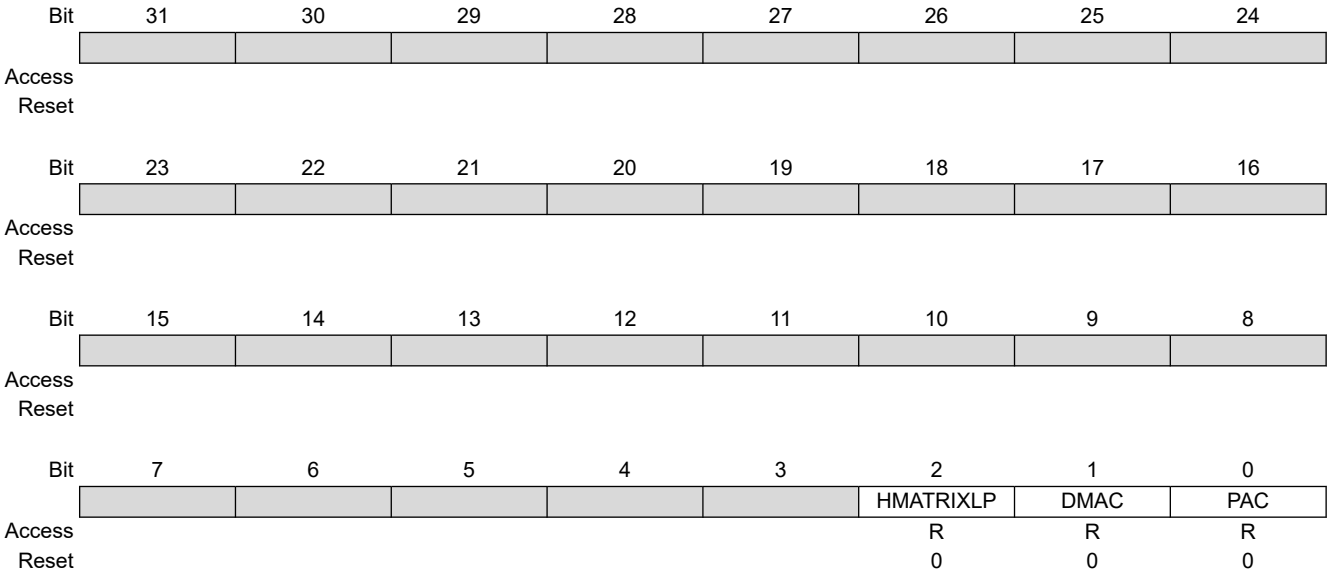
13.1.7.15 Peripheral Write Protection Status E

Name: STATUSE
Offset: 0x44
Reset: 0x000000
Property: –

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.



Bit 2 – HMATRIXLP Peripheral HMATRIXLP Write Protection Status

Bit 1 – DMAC Peripheral DMAC Write Protection Status

Bit 0 – PAC Peripheral PAC Write Protection Status

13.2 Peripherals Configuration Summary

Table 13-3. Peripherals Configuration Summary

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
AHB-APB Bridge A	0x40000000	—	0	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
PM	0x40000000	0	—	—	0	Y	Backup	—	0	N	—	—	—	N/A	PDBACKUP
MCLK	0x40000400	0	—	—	1	Y	Low Power	—	1	N	—	—	—	Y	PD0
RSTC	0x40000800	—	—	—	2	Y	Backup	—	2	N	—	—	—	N/A	PDBACKUP

.....continued

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
OSCCTRL	0x40000C00	0	—	—	3	Y	Low Power	0: DFLL48M reference 1: FDPLL96M clk source 2: FDPLL96M 32kHz	3	N	—	—	—	Y	PD0
OSC32KCTRL	0x40001000	0	—	—	4	Y	Backup	—	4	N	—	—	—	—	PDBACKUP
SUPC	0x40001400	0	—	—	5	Y	Backup	—	5	N	—	—	—	N/A	PDBACKUP
GCLK	0x40001800	—	—	—	6	Y	Low Power	—	6	N	—	—	—	N/A	PD0
WDT	0x40001C00	1	—	—	7	Y	Low Power	—	7	N	—	—	—	Y	PDTOP
RTC	0x40002000	2	—	—	8	Y	Backup	—	8	N	—	1: CMP0/ ALARM0 2: CMP1 3: OVF 4-11: PER0-7	—	Y	PDBACKUP
EIC	0x40002400	3, NMI	—	—	9	Y	Low Power	3	9	N	—	12-27: EXTINT0-15	—	Y	PDTOP
PORT	0x40002800	—	—	—	10	Y	Low Power	—	10	N	0-3 : EV0-3	—	—	Y	PDTOP
AHB-APB Bridge B	0x41000000	—	1	Y	—	—	CPU	—	—	—	—	—	—	N/A	PD2
USB	0x41000000	6	12	Y	0	Y	CPU	4	0	N	—	—	—	Y	PD2
DSU	0x41002000	—	5	Y	1	Y	CPU	—	1	Y	—	—	—	N/A	PD2
NVMCTRL	0x41004000	4	8	Y	2	Y	CPU	—	2	N	—	—	—	Y	PD2
MTB	0x41006000	—	—	—	—	—	CPU	—	—	—	43,44: Start, Stop	—	—	N/A	PD2
AHB-APB Bridge C	0x42000000	—	2	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
SERCOM0	0x42000000	8	—	—	0	Y	Low Power	18: CORE 17: SLOW	0	N	—	—	1: RX 2: TX	Y	PD1
SERCOM1	0x42000400	9	—	—	1	Y	Low Power	19: CORE 17: SLOW	1	N	—	—	3: RX 4: TX	Y	PD1
SERCOM2	0x42000800	10	—	—	2	Y	Low Power	20: CORE 17: SLOW	2	N	—	—	5: RX 6: TX	Y	PD1
SERCOM3	0x42000C00	11	—	—	3	Y	Low Power	21: CORE 17: SLOW	3	N	—	—	7: RX 8: TX	Y	PD1
SERCOM4	0x42001000	12	—	—	4	Y	Low Power	22: CORE 17: SLOW	4	N	—	—	9: RX 10: TX	Y	PD1
TCC0	0x42001400	14	—	—	5	Y	Low Power	25	5	N	12-13: EV0-1 14-17: MC0-3	36: OVF 37: TRG 38: CNT 39-42: MC0-3	11: OVF 12-15: MC0-3	Y	PD1
TCC1	0x42001800	15	—	—	6	Y	Low Power	25	6	N	18-19: EV0-1 20-21: MC0-1	43: OVF 44: TRG 45: CNT 46-47: MC0-1	16: OVF 17-18: MC0-1	Y	PD1
TCC2	0x42001C00	16	—	—	7	Y	Low Power	26	7	N	22-23: EV0-1 24-25: MC0-1	48: OVF 49: TRG 50: CNT 51-52: MC0-1	19: OVF 20-21: MC0-1	Y	PD1

.....continued

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
TC0	0x42002000	17	—	—	8	Y	Low Power	27	8	N	26: EVU	53: OVF 54-55: MC0-1	22: OVF 23-24: MC0-1	Y	PD1
TC1	0x42002400	18	—	—	9	Y	Low Power	27	9	N	27: EVU	56: OVF 57-58: MC0-1	25: OVF 26-27: MC0-1	Y	PD1
RFCTRL	0x42003C00				10	N									
AHB-APB Bridge D	0x43000000	—	3	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
EVSYS	0x43000000	7	—	—	0	Y	Low Power	5-16: one per CHANNEL	0	N	—	—	—	Y	PD0
SERCOM5	0x43000400	13	—	—	1	Y	Low Power	24: CORE 23: SLOW	1	N	—	—	—	Y	PD0
TC4	0x43000800	21	—	—	2	Y	Low Power	29	2	N	—	65: OVF 66-67: MC0-1	34: OVF 35-36: MC0-1	Y	PD0
ADC	0x43000C00	22	—	—	3	Y	Low Power	30	3	N	31: START 32: SYNC	68: RESRDY 69: WINMON	37: RESRDY	Y	PD0
AC	0x43001000	23	—	—	4	Y	Low Power	31	4	N	33-34: SOC0-1	70-71: COMP0-1 72: WIN0	—	Y	PD0
PTC	0x43001400	25	—	—	5	Y	Low Power	33	5	N	37: STCONV	75: EOC 76: WCOMP	—	—	PD0
CCL	0x43001C00	—	—	—	7	Y	Low Power	34	7	N	38 : LUTIN0 39 : LUTIN1 40: LUTIN2 41: LUTIN3	78 : LUTOUT0 79 : LUTOUT1 80: LUTOUT2 81: LUTOUT3	—	Y	PD0
AHB-APB Bridge E	0x44000000	—	4	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
PAC	0x44000000	0	14	Y	0	Y	Low Power	—	0	—	—	82 : ACCERR	—	N/A	PD1
DMAC	0x44000400	—	11	Y	—	—	Low Power	—	1	—	4-11: CH0-7	28-35: CH0-7	—	Y	PD1

13.3 DSU - Device Service Unit

13.3.1 Overview

The Device Service Unit (DSU) provides a means of detecting debugger probes. It enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. For security reasons, some of the DSU features will be limited or unavailable when the device is protected by the NVMCTRL security bit.

Related Links

13.3.11.6 System Services Availability when Accessed Externally and Device is Protected

13.16 NVMCTRL – Non-Volatile Memory Controller

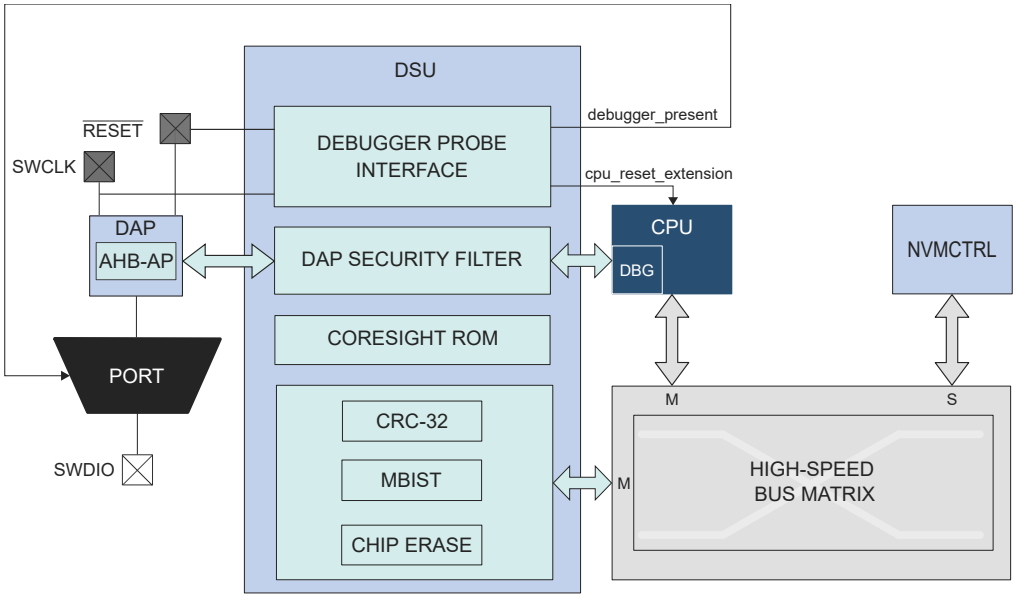
13.16.6.6 Security Bit

13.3.2 Features

- CPU reset extension
- Debugger probe detection (Cold- and Hot-Plugging)
- Chip-Erase command and status
- 32-bit cyclic redundancy check (CRC32) of any memory accessible through the bus matrix
- ARM® CoreSight™ compliant device identification
- Two debug communications channels
- Debug access port security filter
- Onboard memory built-in self-test (MBIST)

13.3.3 Block Diagram

Figure 13-2. DSU Block Diagram



13.3.4 Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

Related Links

7. I/O Multiplexing and Considerations

13.3.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.3.5.1 IO Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU reset phase. For more information, refer to 13.3.6.3 Debugger Probe Detection. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the PORT or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power-reset or an external reset.

13.3.5.2 Power Management

The DSU will continue to operate in any sleep mode where the selected source clock is running.

Related Links

13.8 PM – Power Manager

13.3.5.3 Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Main Clock Controller.

Related Links

13.8 PM – Power Manager

13.6 MCLK – Main Clock

13.6.6.2.6 Peripheral Clock Masking

13.3.5.4 Interrupts

Not applicable.

13.3.5.5 Events

Not applicable.

13.3.5.6 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.3.5.7 Analog Connections

Not applicable.

13.3.6 Debug Operation

13.3.6.1 Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

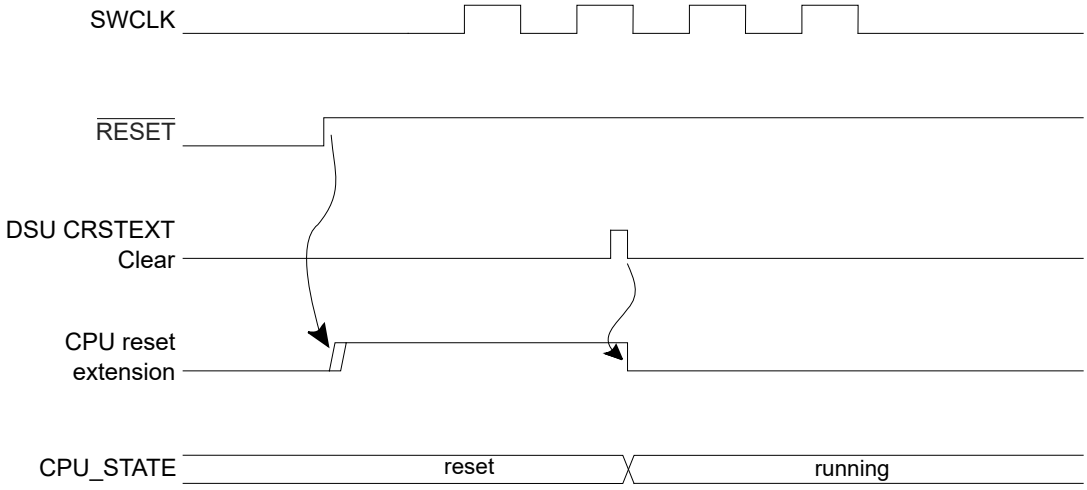
- CPU reset extension
- Debugger probe detection

For more details on the ARM debug components, refer to the ARM Debug Interface v5 Architecture Specification.

13.3.6.2 CPU Reset Extension

“CPU reset extension” refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger is connects to the system. The debugger is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if the SWCLK pin is left unconnected. When the CPU is held in the reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to '0'. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU reset extension when the device is protected by the NVMCTRL security bit. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 13-3. Typical CPU Reset Extension Set and Clear Timing Diagram



Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.6.3 Debugger Probe Detection

13.3.6.3.1 Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in reset. Cold-Plugging is detected when the CPU reset extension is requested, as described above.

13.3.6.3.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in reset. Hot-Plugging is not possible under reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 13-4. Hot-Plugging Detection Timing Diagram

The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the NVMCTRL security bit.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.7 Chip Erase

Chip-Erase consists of removing all sensitive information stored in the chip and clearing the NVMCTRL security bit. Therefore, all volatile memories and the Flash memory (including the EEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must first reset the device in order to be detected. This ensures that internal registers are reset after the protected state is removed. The Chip-Erase operation is triggered by writing a '1' to the Chip-Erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the Chip-Erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The Chip-Erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a Chip- Erase after a Cold-Plugging procedure to ensure that the device is in a known and safe state.

The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (refer to 13.3.6.3.1 Cold Plugging). The device then:
   1.1. Detects the debugger probe.
   1.2. Holds the CPU in reset.
2. Issue the Chip-Erase command by writing a '1' to CTRL.CE. The device then:
   2.1. Clears the system volatile memories.
   2.2. Erases the whole Flash array (including the EEPROM emulation area, not including auxiliary rows).
   2.3. Erases the lock row, removing the NVMCTRL security bit protection.
3. Check for completion by polling STATUSA.DONE (read as '1' when completed).
4. Reset the device to let the NVMCTRL update the fuses.

13.3.8 Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the NVMCTRL security bit. The programming procedure is as follows:

1. At power-up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold (refer to Power-on Reset (POR) characteristics). The system continues to be held in this Static state until the internally regulated supplies have reached a safe Operating state.
2. The PM starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal Resets are maintained due to the external Reset.
3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU remains in Reset due to the Cold-Plugging procedure; meanwhile, the rest of the system is released.
6. A chip erase is issued to ensure that the Flash is fully erased prior to programming.
7. Programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting RESET, toggling power or writing a '1' to the Status A register CPU Reset Phase Extension bit (STATUSA.CRSTEXT). Make sure that the SWCLK pin is high when releasing RESET to prevent extending the CPU reset.

13.3.9 Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and this is accomplished by setting the NVMCTRL security bit. This protected state can be removed by issuing a Chip-Erase (refer to 13.3.7 Chip Erase). When the device is protected, read/write accesses using the AHB-AP are limited to the DSU address range and DSU commands are restricted. When issuing a Chip-Erase, sensitive information is erased from volatile memory and Flash.

The DSU implements a security filter that monitors the AHB transactions inside the DAP. If the device is protected, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (refer to the ARM Debug Interface v5 Architecture Specification on www.arm.com).

The DSU is intended to be accessed either:

- Internally from the CPU, without any limitation, even when the device is protected
- Externally from a debug adapter, with some restrictions when the device is protected

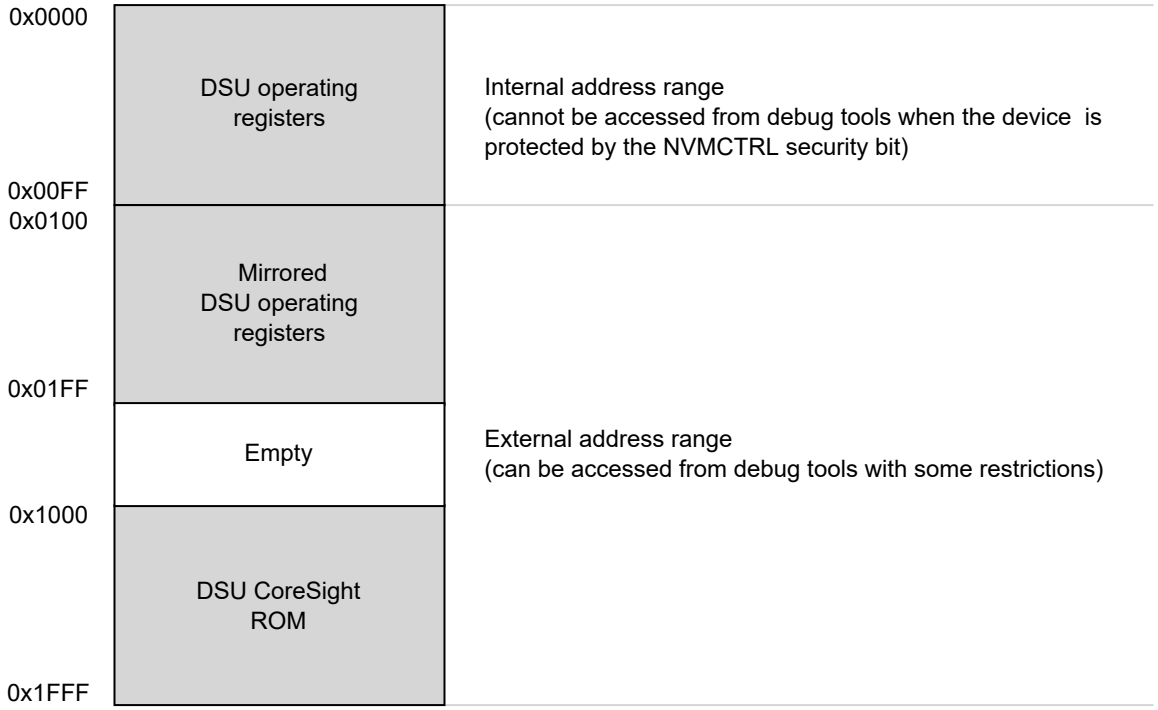
For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map has been mirrored at offset 0x100:

- The first 0x100 bytes form the internal address range
- The next 0x100 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU range 0x0100-0x2000.

The DSU operating registers are located in the 0x0000-0x00FF area and remapped in 0x0100-0x01FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x0100-0x01FF, it is subject to security restrictions. For more information, refer to the Table 13-4.

Figure 13-5. APB Memory Mapping



Some features not activated by APB transactions are not available when the device is protected:

Table 13-4. Feature Availability Under Protection

Features	Availability when the device is protected
CPU Reset Extension	Yes
Clear CPU Reset Extension	No
Debugger Cold-Plugging	Yes
Debugger Hot-Plugging	No

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.10 Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

13.3.10.1 CoreSight Identification

A system-level ARM CoreSight ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 13-6. Conceptual 64-bit Peripheral ID

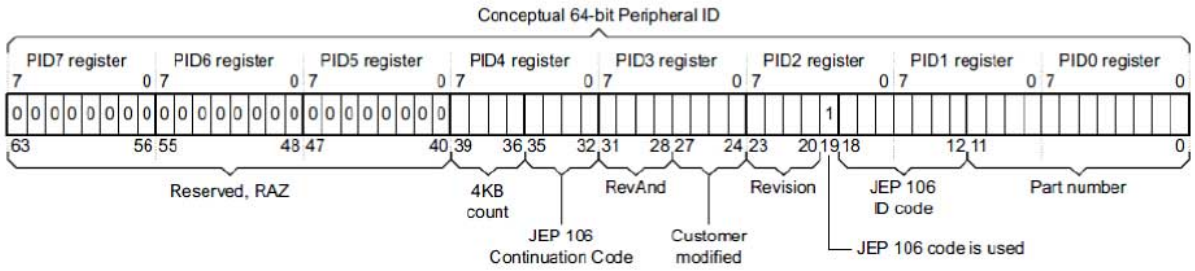


Table 13-5. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Continuation code: 0x0	PID4
JEP-106 ID code	7	Device ID: 0x1F	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
RevAnd	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID)	PID3

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

13.3.10.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

13.3.11 Functional Description

13.3.11.1 Principle of Operation

The DSU provides memory services such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands

are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

13.3.11.2 Basic Operation

13.3.11.2.1 Initialization

The module is enabled by enabling its clocks. For more details, refer to 13.3.5.3 Clocks. The DSU registers can be PAC write-protected.

13.3.11.2.2 Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range 0x100 – 0x2000. If the device is protected by the NVMCTRL security bit, accessing the first 0x100 bytes causes the system to return an error. Refer to 13.3.9 Intellectual Property Protection.

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.11.2.3 Operation From the CPU

There are no restrictions when accessing DSU registers from the CPU. However, the user should access DSU registers in the internal address range (0x0 – 0x100) to avoid external security restrictions. Refer to 13.3.9 Intellectual Property Protection.

13.3.11.3 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is restricted; DATA, ADDR, and LENGTH values are forced (see below)

Table 13-6. AMOD Bit Descriptions when Operating CRC32

AMOD[1:0]	Short name	External range restrictions
0	ARRAY	CRC32 is restricted to the full Flash array area (EEPROM emulation area not included) DATA forced to 0xFFFFFFFF before calculation (no seed)
1	EEPROM	CRC32 of the whole EEPROM emulation area DATA forced to 0xFFFFFFFF before calculation (no seed)
2-3	Reserved	

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

13.3.11.3.1 Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept non-inverted if used as starting point for subsequent CRC32 calculations.

If the device is in protected state by the NVMCTRL security bit, it is only possible to calculate the CRC32 of the whole flash array when operated from the external address space. In most cases, this area will be the entire onboard non-volatile memory. The Address, Length and Data registers will be forced to predefined values once the CRC32 operation is started, and values written by the user are ignored. This allows the user to verify the contents of a protected device.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.11.3.2 Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

13.3.11.4 Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the NVMCTRL security bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.11.5 Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when the device is protected by the NVMCTRL security bit. If an MBIST command is issued when the device is protected, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR).

1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:
   1.1. Write entire memory to '0', in any order.
   1.2. Bit for bit read '0', write '1', in descending order.
   1.3. Bit for bit read '1', write '0', read '0', write '1', in ascending order.
   1.4. Bit for bit read '1', write '0', in ascending order.
   1.5. Bit for bit read '0', write '1', read '1', write '0', in ascending order.
   1.6. Read '0' from entire memory, in ascending order.

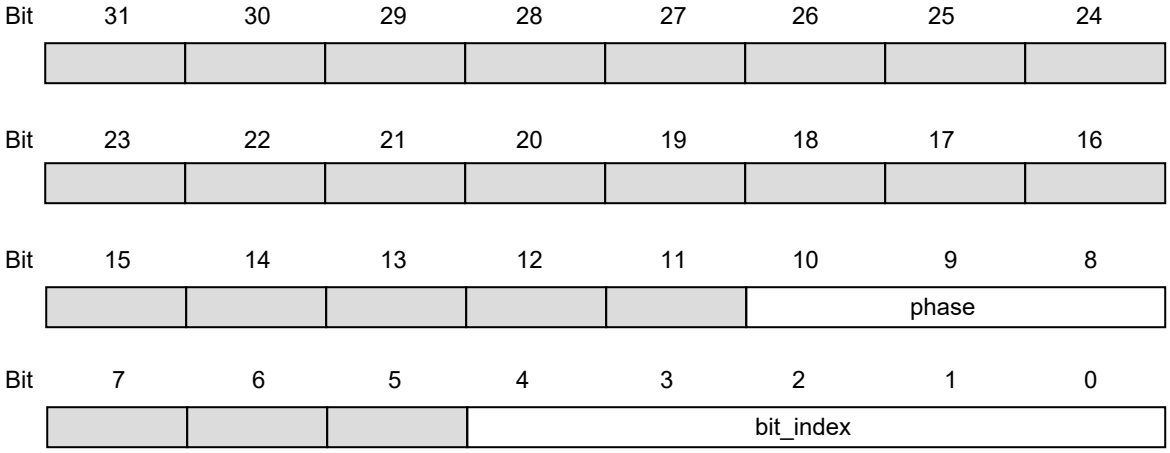
   The specific implementation used has a run time which depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are:
   - Address decoder faults
   - Stuck-at faults
   - Transition faults
   - Coupling faults
   - Linked Coupling faults
2. Starting MBIST
   To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.

For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.

The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.

3. Interpreting the Results
   The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:
   - ADDR.AMOD=0: exit-on-error (default)
     In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
   - ADDR.AMOD=1: pause-on-error
     In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.
4. Locating Faults
   If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   - ADDR: Address of the word containing the failing bit
   - DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 13-7. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 13-7. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address

.....continued	
Phase	**Test actions**
7	Read all zeros. bit_index is not used

Table 13-8. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

Related Links

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.6.6 Security Bit

13.3.11.6 System Services Availability when Accessed Externally and Device is Protected

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x000-0x100 range.

Table 13-9. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range and Device is Protected
Chip-Erase command and status	Yes
CRC32	Yes, only full array or full EEPROM
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
Testing of onboard memories (MBIST)	No
STATUSA.CRSTEXT clearing	No (STATUSA.PERR is set when attempting to do so)

13.3.12 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0				CE	MBIST		CRC	SWRST
0x01	STATUSA	7:0				PERR	FAIL	BERR	CRSTEXT	DONE
0x02	STATUSB	7:0				HPE	DCCD1	DCCD0	DBGPRES	PROT
0x03	Reserved									
0x04	ADDR	7:0	ADDR[5:0]						AMOD[1:0]	
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x08	LENGTH	7:0	LENGTH[5:0]							
		15:8	LENGTH[13:6]							
		23:16	LENGTH[21:14]							
		31:24	LENGTH[29:22]							
0x0C	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x10	DCC0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x14	DCC1	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x18	DID	7:0	DEVSEL[7:0]							
		15:8	DIE[3:0]				REVISION[3:0]			
		23:16	FAMILY[0]		SERIES[5:0]					
		31:24	PROCESSOR[3:0]				FAMILY[4:1]			
0x1C ... 0x0FFF	Reserved									
0x1000	ENTRY0	7:0							FMT	EPRES
		15:8	ADDOFF[3:0]							
		23:16	ADDOFF[11:4]							
		31:24	ADDOFF[19:12]							
0x1004	ENTRY1	7:0							FMT	EPRES
		15:8	ADDOFF[3:0]							
		23:16	ADDOFF[11:4]							
		31:24	ADDOFF[19:12]							
0x1008	END	7:0	END[7:0]							
		15:8	END[15:8]							
		23:16	END[23:16]							
		31:24	END[31:24]							
0x100C ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	7:0								SMEMP
		15:8								
		23:16								
		31:24								
0x1FD0	PID4	7:0	FKBC[3:0]				JEPCC[3:0]			
		15:8								
		23:16								
		31:24								
0x1FD4 ... 0x1FDF	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1FE0	PID0	7:0	PARTNBL[7:0]							
		15:8								
		23:16								
		31:24								
0x1FE4	PID1	7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
		15:8								
		23:16								
		31:24								
0x1FE8	PID2	7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
		15:8								
		23:16								
		31:24								
0x1FEC	PID3	7:0	REVAND[3:0]				CUSMOD[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF0	CID0	7:0	PREAMBLEB0[7:0]							
		15:8								
		23:16								
		31:24								
0x1FF4	CID1	7:0	CCLASS[3:0]				PREAMBLE[3:0]			
		15:8								
		23:16								
		31:24								
0x1FF8	CID2	7:0	PREAMBLEB2[7:0]							
		15:8								
		23:16								
		31:24								
0x1FFC	CID3	7:0	PREAMBLEB3[7:0]							
		15:8								
		23:16								
		31:24								

13.3.13 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.3.5.6 Register Access Protection.

13.3.13.1 Control

Name: CTRL
Offset: 0x0000
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CE	MBIST		CRC	SWRST
Access				W	W		W	W
Reset				0	0		0	0

Bit 4 – CE Chip Erase
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the Chip-Erase operation.

Bit 3 – MBIST Memory Built-In Self-Test
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the memory BIST algorithm.

Bit 1 – CRC 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the cyclic redundancy check algorithm.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the module.

13.3.13.2 Status A

Name: STATUSA
Offset: 0x0001
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				PERR	FAIL	BERR	CRSTEXT	DONE
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – PERR Protection Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Protection Error bit.
This bit is set when a command that is not allowed in protected state is issued.

Bit 3 – FAIL Failure
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Failure bit.
This bit is set when a DSU operation failure is detected.

Bit 2 – BERR Bus Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Bus Error bit.
This bit is set when a bus error is detected.

Bit 1 – CRSTEXT CPU Reset Phase Extension
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CPU Reset Phase Extension bit.
This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU reset phase.

Bit 0 – DONE Done
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Done bit.
This bit is set when a DSU operation is completed.

13.3.13.3 Status B

Name: STATUSB
Offset: 0x0002
Reset: 0x1X
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				HPE	DCCD1	DCCD0	DBGPRES	PROT
Access				R	R	R	R	R
Reset				1	0	0	0	0

Bit 4 – HPE Hot-Plugging Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when Hot-Plugging is enabled.
This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

Bits 2, 3 – DCCDx Debug Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when DCCx is written.
This bit is cleared when DCCx is read.

Bit 1 – DBGPRES Debugger Present
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when a debugger probe is detected.
This bit is never cleared.

Bit 0 – PROT Protected
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set at power-up when the device is protected.
This bit is never cleared.

13.3.13.4 Address

Name: ADDR
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]						AMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:2 – ADDR[29:0] Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0] Access Mode
The functionality of these bits is dependent on the operation mode.
Bit description when operating CRC32: refer to 13.3.11.3 32-bit Cyclic Redundancy Check CRC32
Bit description when testing onboard memories (MBIST): refer to 13.3.11.5 Testing of On-Board Memories MBIST

13.3.13.5 Length

Name: LENGTH
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0] Length
Length in words needed for memory operations.

13.3.13.6 Data

Name: DATA
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Memory operation initial value or result value.

13.3.13.7 Debug Communication Channel 0

Name: DCC0
Offset: 0x0010
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

13.3.13.8 Debug Communication Channel 1

Name: DCC1
Offset: 0x0014
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data
Data register.

13.3.13.9 Device Identification

Name: DID
Offset: 0x0018
Reset: see related links
Property: PAC Write-Protection

The information in this register is related to the Ordering Information.

Bit	31	30	29	28	27	26	25	24
	PROCESSOR[3:0]				FAMILY[4:1]			
Access	R	R	R	R	R	R	R	R
Reset	p	p	p	p	f	f	f	f

Bit	23	22	21	20	19	18	17	16
	FAMILY[0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	f		s	s	s	s	s	s

Bit	15	14	13	12	11	10	9	8
	DIE[3:0]				REVISION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	d	d	d	d	r	r	r	r

Bit	7	6	5	4	3	2	1	0
	DEVSEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – PROCESSOR[3:0] Processor
The value of this field defines the processor used on the device.

Bits 27:23 – FAMILY[4:0] Product Family
The value of this field corresponds to the Product Family part of the ordering code.

Bits 21:16 – SERIES[5:0] Product Series
The value of this field corresponds to the Product Series part of the ordering code.

Bits 15:12 – DIE[3:0] Die Number
Identifies the die family.

Bits 11:8 – REVISION[3:0] Revision Number
Identifies the die revision number. 0x0 = rev.A, 0x1 = rev.B, etc.
Note: The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

Bits 7:0 – DEVSEL[7:0] Device Selection
This bit field identifies a device within a product family and product series. Refer to the Ordering Information for device configurations and corresponding values for Flash memory density, pin count and device variant. For further information, refer to 3.3 Device Identification.

13.3.13.10 CoreSight ROM Table Entry 0

Name: ENTRY0
Offset: 0x1000
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present
This bit indicates whether an entry is present at this location in the ROM table.
This bit is set at power-up if the device is not protected indicating that the entry is not present.
This bit is cleared at power-up if the device is not protected indicating that the entry is present.

13.3.13.11 CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format
Always read as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present
This bit indicates whether an entry is present at this location in the ROM table.
This bit is set at power-up if the device is not protected indicating that the entry is not present.
This bit is cleared at power-up if the device is not protected indicating that the entry is present.

13.3.13.12 CoreSight ROM Table End

Name: END
Offset: 0x1008
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0] End Marker
Indicates the end of the CoreSight ROM table entries.

13.3.13.13 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: 0x0000000X
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SMEMP
Access								R
Reset								x

Bit 0 – SMEMP System Memory Present
This bit indicates whether system memory is present on the bus that connects to the ROM table.
This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

13.3.13.14 Peripheral Identification 4

Name: PID4
Offset: 0x1FD0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0] 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0] JEP-106 Continuation Code
These bits will always return zero when read.

13.3.13.15 Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x000000D0
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PARTNBL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	0	0	0	0

Bits 7:0 – PARTNBL[7:0] Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

13.3.13.16 Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x000000FC
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0] Low part of the JEP-106 Identity Code
These bits will always return 0xF when read (JEP-106 identity code is 0x1F).

Bits 3:0 – PARTNBH[3:0] Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

13.3.13.17 Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x00000009
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bits 7:4 – REVISION[3:0] Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU JEP-106 Identity Code is used
This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0] JEP-106 Identity Code High
These bits will always return 0x1 when read, (JEP-106 identity code is 0x1F).

13.3.13.18 Peripheral Identification 3

Name: PID3
Offset: 0x1FEC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CUSMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0] Revision Number
These bits will always return 0x0 when read.

Bits 3:0 – CUSMOD[3:0] ARM CUSMOD
These bits will always return 0x0 when read.

13.3.13.19 Component Identification 0

Name: CID0
Offset: 0x1FF0
Reset: 0x0000000D
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PREAMBLEB0[7:0] Preamble Byte 0
These bits will always return 0xD when read.

13.3.13.20 Component Identification 1

Name: CID1
Offset: 0x1FF4
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PREAMBLE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0] Component Class
These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (refer to the ARM Debug Interface v5 Architecture Specification at http://www.arm.com).

Bits 3:0 – PREAMBLE[3:0] Preamble
These bits will always return 0x0 when read.

13.3.13.21 Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x00000005
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PREAMBLEB2[7:0] Preamble Byte 2
These bits will always return 0x05 when read.

13.3.13.22 Component Identification 3

Name: CID3
Offset: 0x1FFC
Reset: 0x000000B1
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0] Preamble Byte 3
These bits will always return 0xB1 when read.

13.4 Clock System

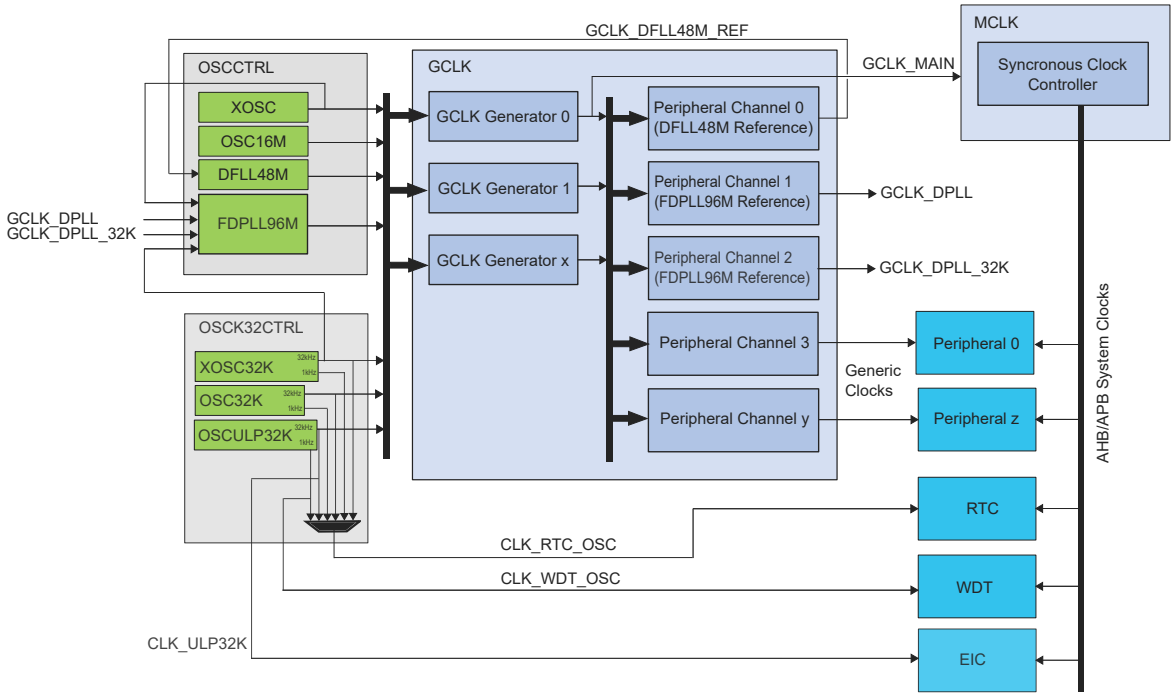
This chapter summarizes the clock distribution and terminology in the SAM R30 device. It will not explain every detail of its configuration. For in-depth documentation, see the respective peripherals descriptions and the *Generic Clock* documentation.

Related Links

13.5 GCLK - Generic Clock Controller
13.6 MCLK – Main Clock

13.4.1 Clock Distribution

Figure 13-8. Clock Distribution

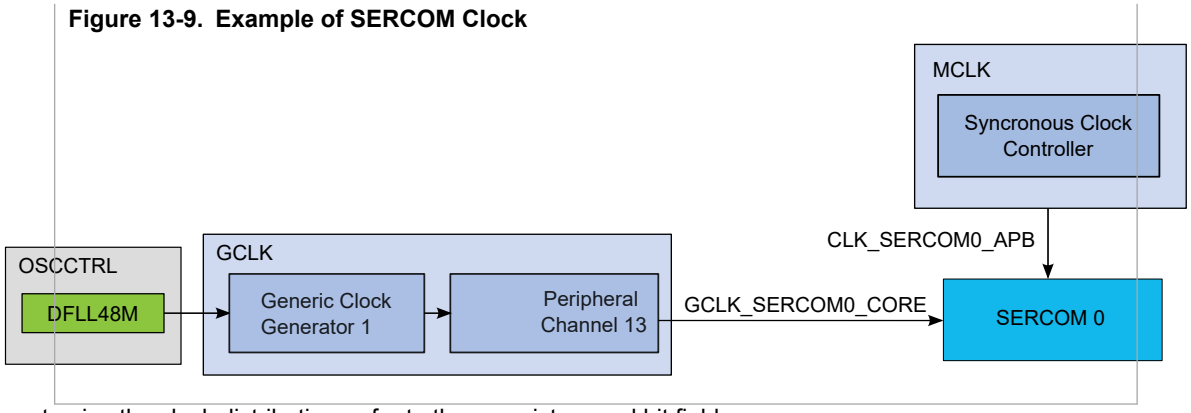


The SAM R30 clock system consists of:

- *Clock sources*, i.e., oscillators controlled by OSCCTRL and OSC32KCTRL
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources are the internal 16MHz oscillator (OSC16M), external crystal oscillator (XOSC) and the Digital Frequency Locked Loop (DFLL48M).
- *Generic Clock Controller (GCLK),* which generates, controls and distributes the asynchronous clock consisting of:
  - *Generic Clock Generators:* These are programmable prescalers that can use any of the system clock sources as a time base. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.
  - *Generic Clocks:* These are clock signals generated by Generic Clock Generators and output by the Peripheral Channels, and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock for each instance. Generic Clock 0 serves as the clock source for the DFLL48M clock input (when multiplying another clock source).
- *Main Clock Controller (MCLK)*
  - The MCLK generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB) as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The next figure shows an example where SERCOM0 is clocked by the DFLL48M in open loop mode. The DFLL48M is enabled, the Generic Clock Generator 1 uses the DFLL48M as its clock source and feeds into Peripheral Channel 13. The Generic Clock 13, also called GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, was unmasked in the APBC Mask register in the MCLK.

Figure 13-9. Example of SERCOM Clock



To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a generic clock generator, n, is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel, m, can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN).
- The Peripheral Channel number, m, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm.
- The AHB clocks are enabled and disabled by writing to the respective bit in the AHB Mask register (MCLK.AHBMASK).
- The APB clocks are enabled and disabled by writing to the respective bit in the APB x Mask registers (MCLK.APBxMASK).

13.4.2 Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, i.e. they are clocked from different clock sources and/or with different clock speeds, some peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.

For a general description, see 13.4.3 Register Synchronization. Some peripherals have specific properties described in their individual sub-chapter “Synchronization”.

In the datasheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

Related Links

13.5.6.6 Synchronization

13.4.3 Register Synchronization

13.4.3.1 Overview

All peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

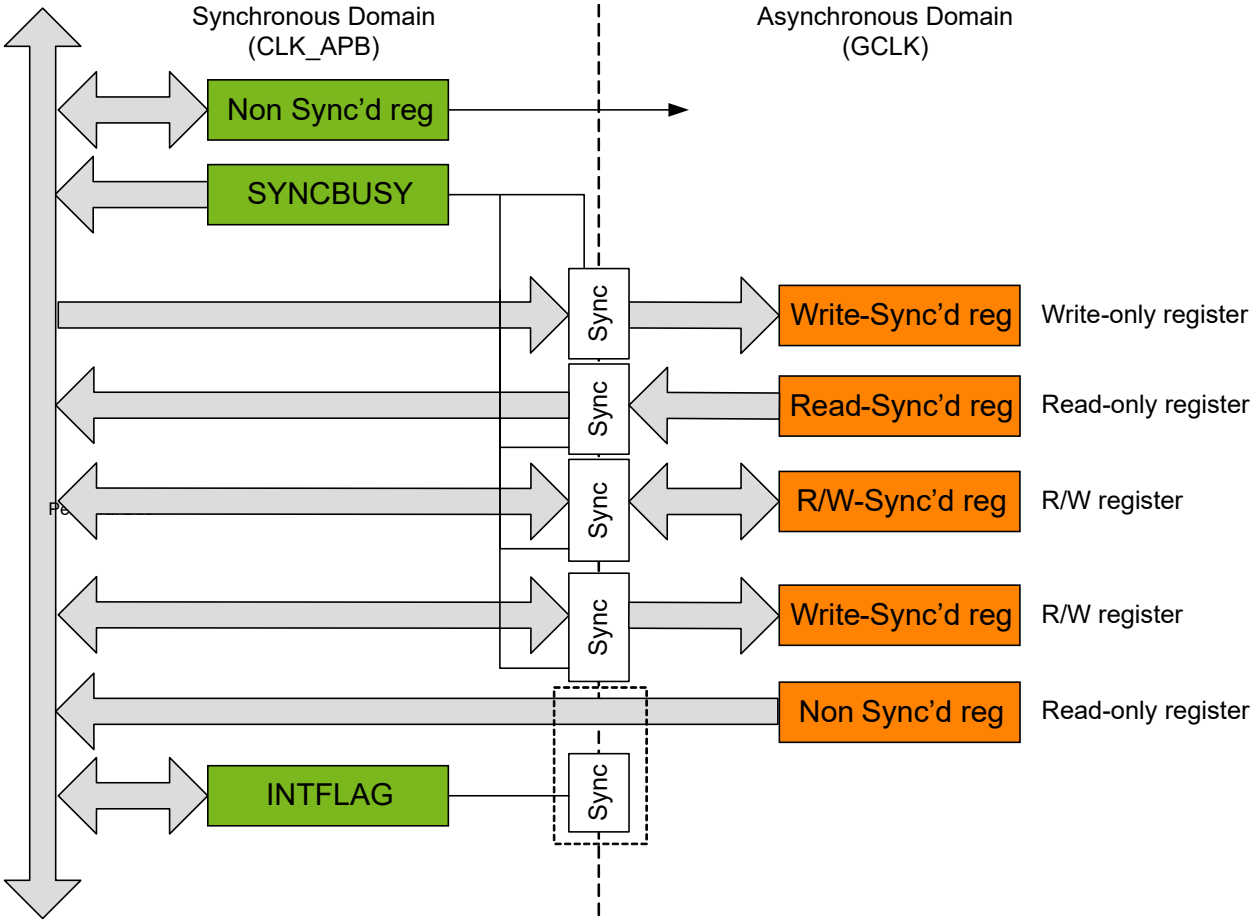
All registers in the bus interface are accessible without synchronization.

All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).
Note: For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

Figure 13-10. Register Synchronization Overview



13.4.3.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain. The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer to 13.4.3.7 Synchronization Delay for details on the synchronization delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported.

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

13.4.3.3 General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode. Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

13.4.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

13.4.3.5 Enable Write Synchronization

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronisation Ready interrupt (if available) cannot be used to enable write-synchronization.

13.4.3.6 Software Reset Write-Synchronization

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a ‘1’ to the CTRLA.SWRST bit it will immediately read as ‘1’.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

13.4.3.7 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

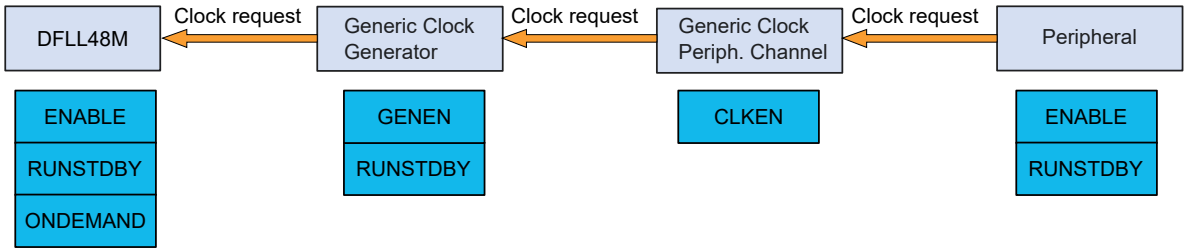
13.4.4 Enabling a Peripheral

In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.
- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled.
- The user interface of the peripheral needs to be unmasked in the PM. If this is not done the peripheral registers will read all 0’s and any writing attempts to the peripheral will be discarded.

13.4.5 On Demand Clock Requests

Figure 13-11. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, via the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

T_{start_max} = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

T_{start_min} = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, T_{stop}, is between:

T_{stop_min} = 1 × divided clock source period + 1 × clock source period

T_{stop_max} = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules, see Figure 13-11.

13.4.6 Power Consumption vs. Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

13.4.7 Clocks after Reset

On any Reset, the synchronous clocks start to their initial state:

- OSC16M is enabled and configured to run at 4 MHz
- Generic Generator 0 uses OSC16M as source and generates GCLK_MAIN
- CPU and BUS clocks are undivided

On a Power Reset, the 32 KHz clock sources are reset, and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except
  - Generator 0 is using OSC16M at 4 MHz as source and generates GCLK_MAIN
- All Peripheral Channels in GCLK are disabled

On a User Reset, the GCLK module starts in its initial state, except for:

- Generic Clocks that are write-locked, i.e., the according WRTLOCK is set to 1 prior to Reset

13.5 GCLK - Generic Clock Controller

13.5.1 Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller GCLK provides nine Generic Clock Generators [8:0] that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 13-13. The number of Peripheral Clocks depends on how many peripherals the device has.

Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

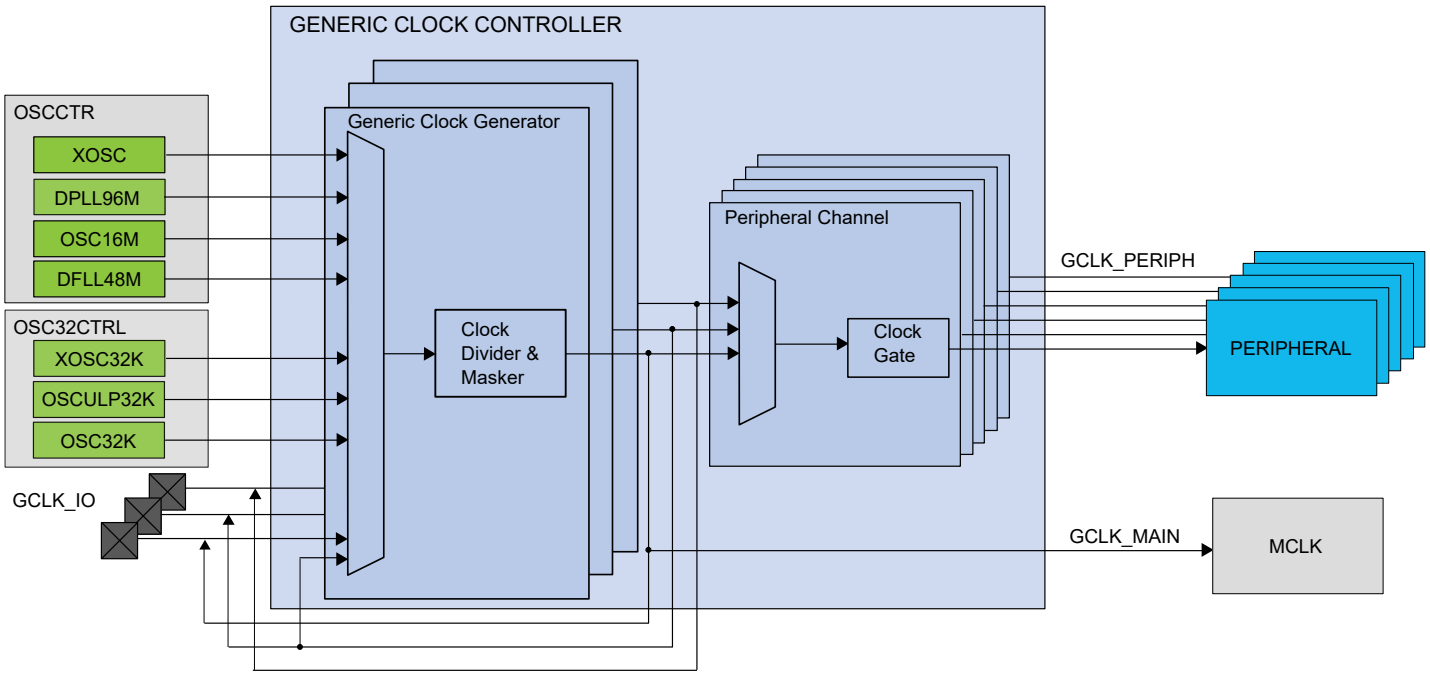
13.5.2 Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range

13.5.3 Block Diagram

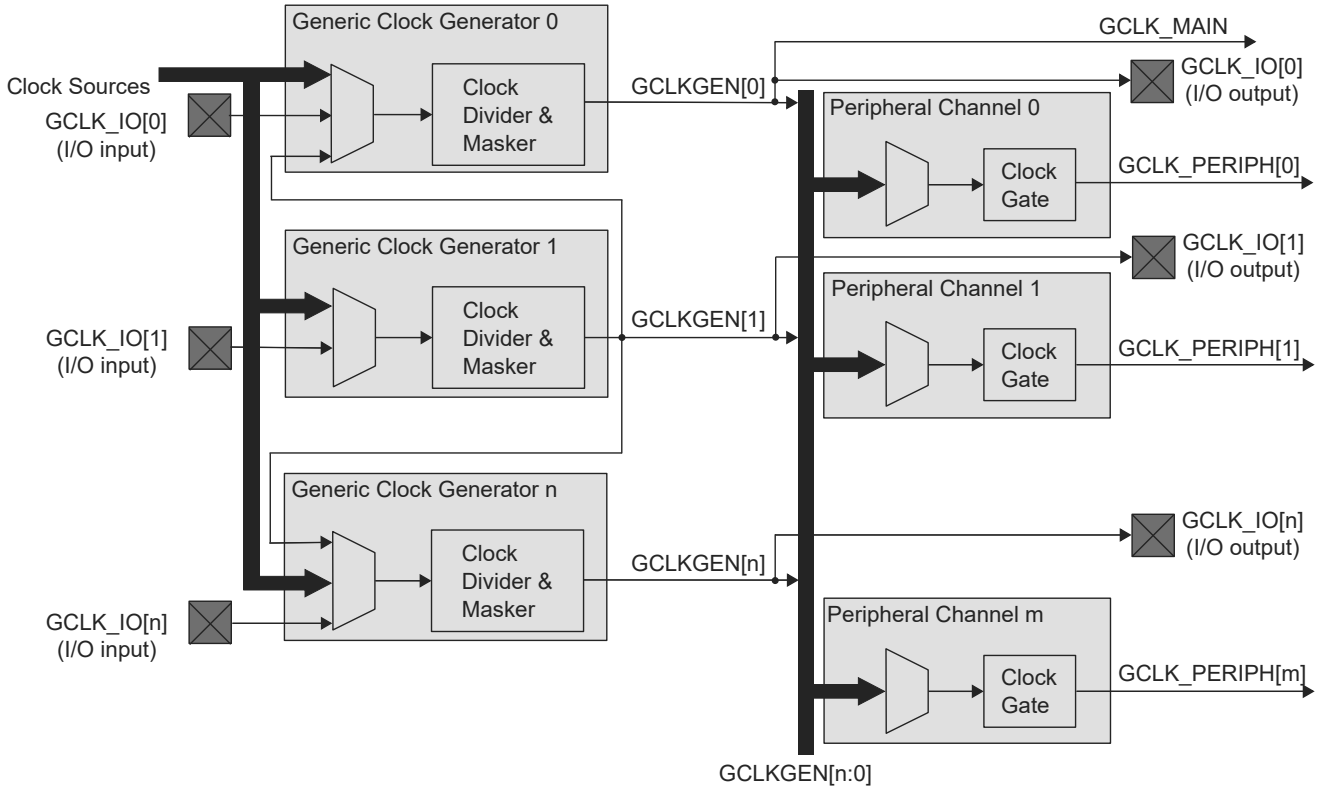
The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

Figure 13-12. Device Clocking Diagram



The GCLK block diagram is shown below:

Figure 13-13. Generic Clock Controller Block Diagram



13.5.4 Signal Description

Table 13-10. GCLK Signal Description

Signal Name	Type	Description
GCLK_IO[7:0]	Digital I/O	Clock source for Generators when input Generic Clock signal when output

Note: One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.5.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.5.5.1 I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

Related Links

13.17 PORT - I/O Pin Controller

13.5.5.2 Power Management

The GCLK can operate in all sleep modes, if required.

Related Links

13.8 PM – Power Manager

13.5.5.3 Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.5.5.4 DMA

Not applicable.

13.5.5.5 Interrupts

Not applicable.

13.5.5.6 Events

Not applicable.

13.5.5.7 Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.5.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.5.5.9 Analog Connections

Not applicable.

13.5.6 Functional Description

13.5.6.1 Principle of Operation

The GCLK module is comprised of nine Generic Clock Generators (Generators) sourcing up to 64 Peripheral Channels and the Main Clock signal GCLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

13.5.6.2 Basic Operation

13.5.6.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator n is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.

13.5.6.2.2 Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to 13.5.6.3.4 Configuration Lock.

13.5.6.2.3 Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of nine different clock sources except GCLK_GEN[1], which can be set to run from one of eight sources. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

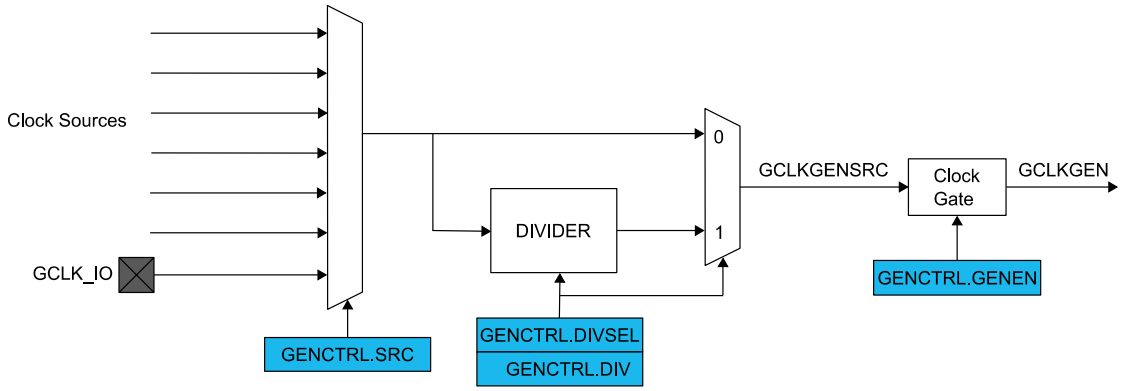
Each generator GCLK_GEN[x] can be connected to one specific pin (GCLK_IO[y]). The GCLK_IO[y] can be set to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

Figure 13-14. Generic Clock Generator



Related Links

13.6 MCLK – Main Clock

13.5.6.2.4 Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

13.5.6.2.5 Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated.

13.5.6.2.6 Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL, and DFLL). Only Generator 1 can be used as a common source for all other generators.

13.5.6.2.7 Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The number of DIV bits for each Generator is device dependent.

13.5.6.2.8 Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

13.5.6.2.9 External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.

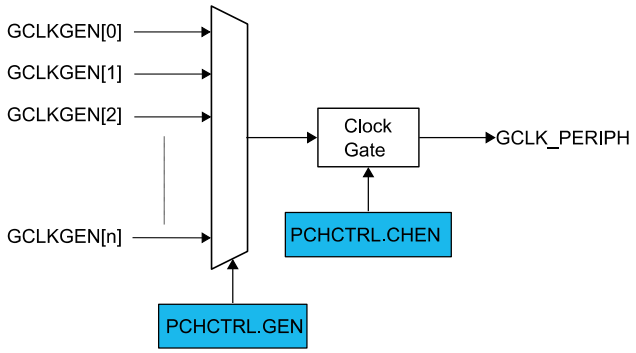
If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low when turned off. If this bit is '1', the output clock will be high when turned off.

In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero.
Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay between turning off Power Domain 1 and 2. The maximum delay can be equal to the clock source period multiplied by the division factor.

If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

13.5.6.3 Peripheral Clock

Figure 13-15. Peripheral Clock



13.5.6.3.1 Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRL.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

13.5.6.3.2 Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

13.5.6.3.3 Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN=0
2. Assert that PCHCTRLm.CHEN reads '0'
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN=1

13.5.6.3.4 Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

There is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

Related Links

13.5.8.1 CTRLA

13.5.6.4 Additional Features

13.5.6.4.1 Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

13.5.6.5 Sleep Mode Operation

13.5.6.5.1 SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to 13.5.6.2.9 External Clock for details.

Related Links

13.8 PM – Power Manager

13.5.6.5.2 Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

Table 13-11. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
yes	-	-	active
no	1	1	active
no	1	0	OFF
no	0	1	OFF
no	0	0	OFF

13.5.6.5.3 Entering Standby Mode

There may occur a delay when the device is put into Standby, until the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

Related Links

13.8 PM – Power Manager

13.5.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

13.5.8.1 CTRLA

13.5.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SYNCBUSY	7:0	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
		15:8						GENCTRL8	GENCTRL7	GENCTRL6
		23:16								
		31:24								
0x08 ... 0x1F	Reserved									
0x20	GENCTRL0	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x24	GENCTRL1	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x28	GENCTRL2	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x2C	GENCTRL3	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x30	GENCTRL4	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x34	GENCTRL5	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x38	GENCTRL6	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x3C	GENCTRL7	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x40	GENCTRL8	7:0				SRC[4:0]				
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		23:16	DIV[7:0]							
		31:24	DIV[15:8]							
0x44 ... 0x7F	Reserved									
0x80	PCHCTRL0	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								
...										

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0108	PCHCTRL34	7:0	WRTLOCK	CHEN				GEN[2:0]		
		15:8								
		23:16								
		31:24								

13.5.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.5.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.5.6.6 Synchronization.

13.5.8.1 Control A

Name: CTRLA
Offset: 0x0
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								R/W
Reset								0

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.
Refer to GENCTRL Reset Value for details on GENCTRL register reset.
Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.
Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset.
CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

13.5.8.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: —

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						GENCTRL8	GENCTRL7	GENCTRL6
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bits 2, 3, 4, 5, 6, 7, 8, 9, 10 – GENCTRLn Generator Control n Synchronization Busy
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete or when clock switching operation is complete.
This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

13.5.8.3 Generator Control

Name: GENCTRLn
Offset: 0x20 + n*0x04 [n=0..8]
Reset: 0x00000106 (GENCTRL0), 0x00000000 (others)
Property: PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=8..0). The reset value is 0x00000106 (GENCTRL0), 0x00000000 (others).

Bit	31	30	29	28	27	26	25	24
	DIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SRC[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 31:16 – DIV[15:0] Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor used is dependent on the state of DIVSEL. The number of relevant DIV bits for each Generator can be seen in this table. Written bits outside of the specified range will be ignored.

Table 13-12. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Generator 0	8 division factor bits – DIV[7:0]
Generator 1	16 division factor bits – DIV[15:0]
Generator 2-8	8 division factor bits – DIV[7:0]

Bit 13 – RUNSTDBY Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO pin during Standby mode.

Bit 12 – DIVSEL Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either '0' or '1'.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(GENCTRLn.DIV+1).

Bit 11 – OE Output Enable
This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field. This feature only applies to GCLK Clock Generators 0 through 7. (GCLK Generator 8 does not have a GCLK_IO pin.)

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV Output Off Value
This bit is used to control the clock output value on pin (GCLK_IO) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field. This feature only applies to GCLK Clock Generators 0 through 7. (GCLK Generator 8 does not have a GCLK_IO pin.)

Value	Description
0	The GCLK_IO will be LOW when the generator is turned off or when the OE bit is zero.
1	The GCLK_IO will be HIGH when the generator is turned off or when the OE bit is zero.

Bit 9 – IDC Improve Duty Cycle
This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN Generator Enable
This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 4:0 – SRC[4:0] Generator Clock Source Selection
These bits select the Generator clock source, as shown in this table.

Table 13-13. Generator Clock Source Selection

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLK_IN	Generator input pad (GCLK_IO)
0x02	GCLK_GEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	OSC32K	OSC32K oscillator output
0x05	XOSC32K	XOSC32K oscillator output
0x06	OSC16M	OSC16M oscillator output
0x07	DFLL48M	DFLL48M output
0x08	DPLL96M	DPLL96M output
0x09-0x1F	Reserved	Reserved for future use

A Power Reset will reset all GENCTRLn registers. the Reset values of the GENCTRLn registers are shown in the table below.

Table 13-14. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
0	0x00000106
Others	0x00000000

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK=1). The reset values of the GENCTRL register are as shown in the table below.

Table 13-15. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
0	0x00000106
Others	No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK = 1 Else 0x00000000

13.5.8.4 Peripheral Channel Control

Name: PCHCTRLm
Offset: 0x80 + m*0x04 [m=0..34]
Reset: 0x00000000
Property: PAC Write-Protection

PCHTRLm controls the settings of Peripheral Channel number m (m=0..34).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN				GEN[2:0]		
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit 7 – WRTLOCK Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.
Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN Channel Enable
This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 2:0 – GEN[2:0] Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Table 13-16. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3
0x4	Generic Clock Generator 4
0x5	Generic Clock Generator 5
0x6	Generic Clock Generator 6
0x7	Generic Clock Generator 7

.....continued	
Value	**Description**
0x8	Generic Clock Generator 8
0x9 – 0xF	Reserved

Table 13-17. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	No change

A Power Reset will reset all the PCHCTRLm registers.
A User Reset will reset a PCHCTRL if WRTLOCK = 0 or else, the content of that PCHCTRL remains unchanged.
The PCHCTRL register Reset values are shown in the table below, PCHCTRLm Mapping.

Table 13-18. PCHCTRLm Mapping

index(m)	Name	Description
0	GCLK_DFLL48M_REF	DFLL48M Reference
1	GCLK_DPLL	FDPLL96M input clock source for reference
2	GCLK_DPLL_32K	FDPLL96M 32kHz clock for FDPLL96M internal lock timer
3	GCLK_EIC	EIC
4	GCLK_USB	USB
5	GCLK_EVSYS_CHANNEL_0	EVSYS_CHANNEL_0
6	GCLK_EVSYS_CHANNEL_1	EVSYS_CHANNEL_1
7	GCLK_EVSYS_CHANNEL_2	EVSYS_CHANNEL_2
8	GCLK_EVSYS_CHANNEL_3	EVSYS_CHANNEL_3
9	GCLK_EVSYS_CHANNEL_4	EVSYS_CHANNEL_4
10	GCLK_EVSYS_CHANNEL_5	EVSYS_CHANNEL_5
11	GCLK_EVSYS_CHANNEL_6	EVSYS_CHANNEL_6
12	GCLK_EVSYS_CHANNEL_7	EVSYS_CHANNEL_7
13	GCLK_EVSYS_CHANNEL_8	EVSYS_CHANNEL_8
14	GCLK_EVSYS_CHANNEL_9	EVSYS_CHANNEL_9
15	GCLK_EVSYS_CHANNEL_10	EVSYS_CHANNEL_10
16	GCLK_EVSYS_CHANNEL_11	EVSYS_CHANNEL_11
17	GCLK_SERCOM[0,1,2,3,4]_SLOW	SERCOM[0,1,2,3,4]_SLOW
18	GCLK_SERCOM0_CORE	SERCOM0_CORE
19	GCLK_SERCOM1_CORE	SERCOM1_CORE
20	GCLK_SERCOM2_CORE	SERCOM2_CORE
21	GCLK_SERCOM3_CORE	SERCOM3_CORE
22	GCLK_SERCOM4_CORE	SERCOM4_CORE
23	GCLK_SERCOM5_SLOW	SERCOM5_SLOW
24	GCLK_SERCOM5_CORE	SERCOM5_CORE
25	GCLK_TCC0, GCLK_TCC1	TCC0, TCC1
26	GCLK_TCC2	TCC2
27	GCLK_TC0, GCLK_TC1	TC0, TC1
28	—	—
29	GCLK_TC4	TC4
30	GCLK_ADC	ADC
31	GCLK_AC	AC
32	—	—
33	GCLK_PTC	PTC
34	GCLK_CCL	CCL

13.6 MCLK – Main Clock

13.6.1 Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

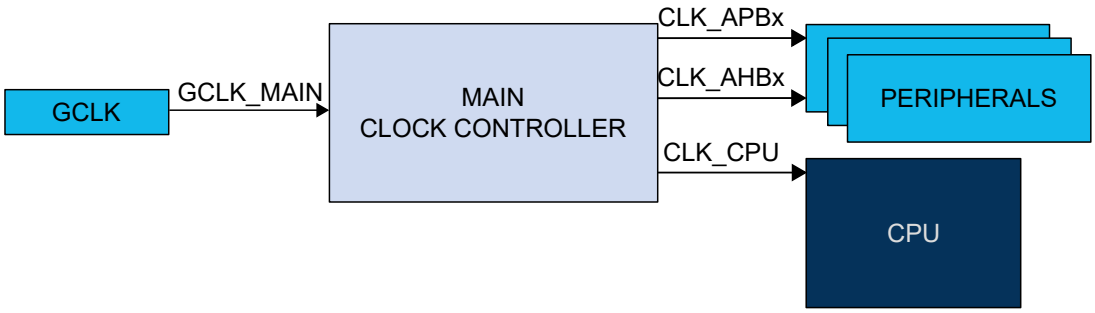
Using a clock provided by the Generic Clock Module (GCLK_MAIN), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx bus. The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

13.6.2 Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

13.6.3 Block Diagram

Figure 13-16. MCLK Block Diagram



13.6.4 Signal Description

Not applicable.

13.6.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.6.5.1 I/O Lines

Not applicable.

13.6.5.2 Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.

Related Links

13.8 PM – Power Manager

13.6.5.3 Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed.

Related Links

13.5 GCLK - Generic Clock Controller

13.6.6.2.6 Peripheral Clock Masking

13.6.5.3.1 Main Clock

The main clock GCLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

13.6.5.3.2 CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

13.6.5.3.3 APBx and AHBx Clock

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock, but can be divided by a prescaler, and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

13.6.5.3.4 Clock Domains

The device has these synchronous clock domains:

- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.
- Low Power synchronous clock domain (LP Clock Domain). Frequency is f_{LP}.
- Backup synchronous clock domain. (BUP Clock Domain). Frequency is f_{BUP}.

See the references for the clock domain partitioning. Refer to the *Maximum Clock Frequencies* in the *Electrical Characteristics* chapter for the maximum frequencies in each performance level.

13.6.5.4 DMA

Not applicable.

13.6.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

13.6.5.6 Events

Not applicable.

13.6.5.7 Debug Operation

When the CPU is halted in debug mode, the MCLK continues normal operation. In sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in debug mode.

13.6.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.6.5.9 Analog Connections

Not applicable.

13.6.6 Functional Description

13.6.6.1 Principle of Operation

The GCLK_MAIN clock signal from the GCLK module is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The GCLK_MAIN is divided by an 8-bit

prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. Each clock domain (CPU, LP, BUP) can be changed on the fly to respond to variable load in the application as long as $f_{CPU} \geq f_{LP} \geq f_{BUP}$. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

13.6.6.2 Basic Operation

13.6.6.2.1 Initialization

After a Reset, the default clock source of the GCLK_MAIN clock is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.6.6.2.2 Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

13.6.6.2.3 Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

Related Links

13.5 GCLK - Generic Clock Controller

13.6.6.2.4 Selecting the Synchronous Clock Division Ratio

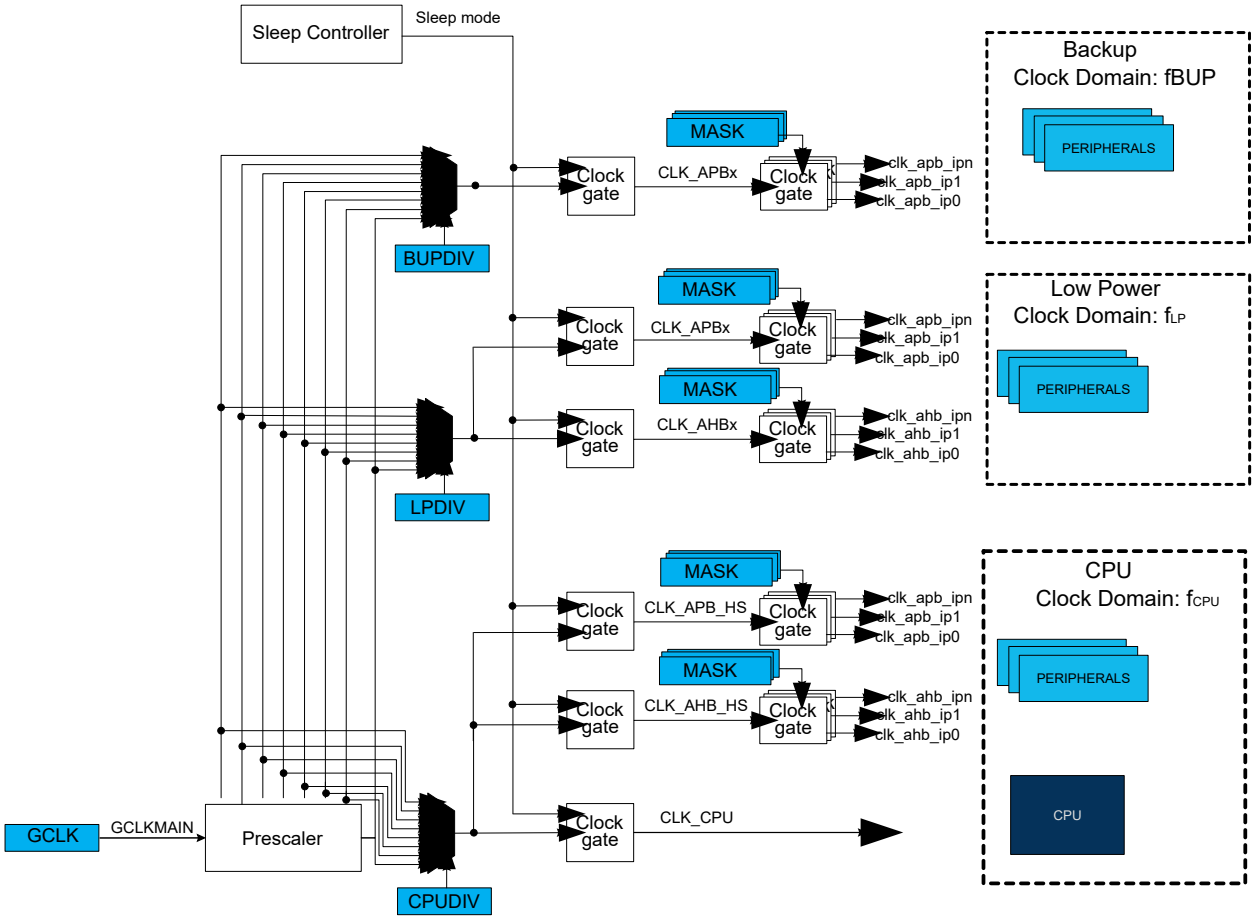
The main clock, CLK_MAIN, feeds an 8-bit prescaler that can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

If the application attempts to write forbidden values in the CPUDIV and LPDIV registers, the registers are written but these bad values are not used, and a violation is reported to the PAC module.

Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time.

Figure 13-17. Synchronous Clock Selection and Prescaler



References:

15. Electrical Characteristics

13.1 PAC - Peripheral Access Controller

13.6.6.2.5 Clock Ready Flag

There is a slight delay between writing to CPUDIV, LPDIV, and BUPDIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The clock settings (CLKCFG) must not be re-written while INTFLAG. CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

13.6.6.2.6 Peripheral Clock Masking

It is possible to disable/enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers (APBxMASK) to '0'/'1'. The default state of the peripheral clocks is shown here.

Table 13-19. Peripheral Clock Default State

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_BRIDGE_B_AHB	Enabled
CLK_DSU_AHB	Enabled
CLK_DSU_APB	Enabled

.....continued

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_USB_AHB	Enabled
CLK_USB_APB	Enabled
CLK_NVMCTRL_AHB	Enabled
CLK_NVMCTRL_APB	Enabled

Backup Clock Domain	
Peripheral Clock	**Default State**
CLK_OSC32KCTRL_APB	Enabled
CLK_PM_APB	Enabled
CLK_SUPC_APB	Enabled
CLK_RSTC_APB	Enabled
CLK_RTC_APB	Enabled

Low Power Clock Domain	
Peripheral Clock	**Default State**
CLK_AC_APB	Enabled
CLK_ADC_APB	Enabled
CLK_BRIDGE_A_AHB	Enabled
CLK_BRIDGE_C_AHB	Enabled
CLK_BRIDGE_D_AHB	Enabled
CLK_BRIDGE_E_AHB	Enabled
CLK_CCL_APB	Enabled
CLK_DMAC_AHB	Enabled
CLK_EIC_APB	Enabled
CLK_EVSYS_APB	Enabled
CLK_GCLK_APB	Enabled
CLK_MCLK_APB	Enabled
CLK_OSCCTRL_APB	Enabled
CLK_PAC_AHB	Enabled
CLK_PAC_APB	Enabled
CLK_PORT_APB	Enabled
CLK_PTC_APB	Enabled
CLK_SERCOM0_APB	Enabled
CLK_SERCOM1_APB	Enabled
CLK_SERCOM2_APB	Enabled

.....continued	
Low Power Clock Domain	
Peripheral Clock	**Default State**
CLK_SERCOM3_APB	Enabled
CLK_SERCOM4_APB	Enabled
CLK_SERCOM5_APB	Enabled
CLK_TCC0_APB	Enabled
CLK_TCC1_APB	Enabled
CLK_TCC2_APB	Enabled
CLK_TC0_APB	Enabled
CLK_TC1_APB	Enabled
CLK_TC4_APB	Enabled
CLK_WDT_APB	Enabled

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

Note that clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

13.6.6.3 DMA Operation

Not applicable.

13.6.6.4 Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU, LP, and BUP clocks are ready. This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

Related Links

11.2.1 Overview
13.8 PM – Power Manager

13.6.6.5 Events

Not applicable.

13.6.6.6 Sleep Mode Operation

In IDLE sleep mode, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.

13.6.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	CPUDIV	7:0	CPUDIV[7:0]							
0x05	LPDIV	7:0	LPDIV[7:0]							
0x06	BUPDIV	7:0	BUPDIV[7:0]							
0x07 ... 0x0F	Reserved									
0x10	AHBMASK	7:0	Reserved	Reserved	DSU	APBE	APBD	APBC	APBB	APBA
		15:8		PAC	Reserved	USB	DMAC	Reserved	Reserved	NVMCTRL
		23:16								
		31:24								
0x14	APBAMASK	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
		15:8	Reserved[3:0]					PORT	EIC	RTC
		23:16	Reserved[11:4]							
		31:24	Reserved[19:12]							
0x18	APBBMASK	7:0	Reserved[4:0]					NVMCTRL	DSU	USB
		15:8	Reserved[12:5]							
		23:16	Reserved[20:13]							
		31:24	Reserved[28:21]							
0x1C	APBCMASK	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
		15:8	RFCTRL						TC1	TC0
		23:16								
		31:24								
0x20	APBDMASK	7:0	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
		15:8								
		23:16								
		31:24								
0x24	APBEMASK	7:0	Reserved[6:0]							PAC
		15:8	Reserved[14:7]							
		23:16	Reserved[22:15]							
		31:24	Reserved[30:23]							

13.6.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the 13.6.5.8 Register Access Protection for details.

13.6.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

All bits in this register are reserved.

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

13.6.8.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled and will generate an interrupt request when the Clock Ready Interrupt Flag is set.

13.6.8.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

13.6.8.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x03
Reset: 0x01
Property: –

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								1

Bit 0 – CKRDY Clock Ready
This flag is cleared by writing a '1' to the flag.
This flag is set when the synchronous CPU, APBx, and AHBx clocks have frequencies as indicated in the CLKCFG registers and will generate an interrupt if INTENCLR/SET.CKRDY is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Clock Ready interrupt flag.

13.6.8.5 CPU Clock Division

Name: CPUDIV
Offset: 0x04
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	CPUDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – CPUDIV[7:0] CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain. Frequencies must never exceed the specified maximum frequency for each clock domain. To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP}$ (i.e., LPDIV ≥CPUDIV). Frequencies must never exceed the specified maximum frequency for each clock domain.
Refer to the Maximum Clock Frequencies in the Electrical Characterization section for maximum frequencies in each performance level.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	—	Reserved

13.6.8.6 Low Power Clock Division

Name: LPDIV
Offset: 0x05
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	LPDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – LPDIV[7:0] Low-Power Clock Division Factor
These bits define the division ratio of the main clock prescaler (2^n) related to the Low Power clock domain. To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP} \geq F\_{BUP}$ (i.e., BUPDIV ≥ LPDIV ≥ CPUDIV). Also, frequencies must never exceed the specified maximum frequency for each clock domain.
Refer to the Maximum Clock Frequencies in the Electrical Characterization section for maximum frequencies in each performance level.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	—	Reserved

13.6.8.7 Backup Clock Division

Name: BUPDIV
Offset: 0x06
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BUPDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BUPDIV[7:0] Backup Clock Division Factor
These bits define the division ratio of the main clock prescaler (2^n) related to the Backup clock domain. To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP} \geq F\_{BUP}$ (i.e. BUPDIV ≥ LPDIV ≥ CPUDIV). Also, frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

13.6.8.8 AHB Mask

Name: AHBMASK
Offset: 0x10
Reset: 0x000FFFFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		PAC	Reserved	USB	DMAC	Reserved	Reserved	NVMCTRL
Access		R	R	R/W	R/W	R	R	R/W
Reset		1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	DSU	APBE	APBD	APBC	APBB	APBA
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 14 – PAC PAC AHB Clock Enable

Value	Description
0	The AHB clock for the PAC is stopped.
1	The AHB clock for the PAC is enabled.

Bits 13,10,9,7,6 – Reserved Reserved bits
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 12 – USB USB AHB Clock Enable

Value	Description
0	The AHB clock for the USB is stopped.
1	The AHB clock for the USB is enabled.

Bit 11 – DMAC DMAC AHB Clock Enable

Value	Description
0	The AHB clock for the DMAC is stopped.
1	The AHB clock for the DMAC is enabled.

Bit 8 – NVMCTRL NVMCTRL AHB Clock Enable

Value	Description
0	The AHB clock for the NVMCTRL is stopped.
1	The AHB clock for the NVMCTRL is enabled.

Bit 5 – DSU DSU AHB Clock Enable

Value	Description
0	The AHB clock for the DSU is stopped.
1	The AHB clock for the DSU is enabled.

Bit 4 – APBE APBE AHB Clock Enable

Value	Description
0	The AHB clock for the APBE is stopped.
1	The AHB clock for the APBE is enabled.

Bit 3 – APBD APBD AHB Clock Enable

Value	Description
0	The AHB clock for the APBD is stopped.
1	The AHB clock for the APBD is enabled

Bit 2 – APBC APBC AHB Clock Enable

Value	Description
0	The AHB clock for the APBC is stopped.
1	The AHB clock for the APBC is enabled

Bit 1 – APBB APBB AHB Clock Enable

Value	Description
0	The AHB clock for the APBB is stopped.
1	The AHB clock for the APBB is enabled.

Bit 0 – APBA APBA AHB Clock Enable

Value	Description
0	The AHB clock for the APBA is stopped.
1	The AHB clock for the APBA is enabled.

13.6.8.9 APBA Mask

Name: APBAMASK
Offset: 0x14
Reset: 0x00001FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[3:0]					PORT	EIC	RTC
Access	R	R	R	R		R	R	R
Reset	0	0	0	1		1	1	1

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:12 – Reserved[19:0] For future use
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 10 – PORT PORT APBA Clock Enable

Value	Description
0	The APBA clock for the PORT is stopped.
1	The APBA clock for the PORT is enabled.

Bit 9 – EIC EIC APBA Clock Enable

Value	Description
0	The APBA clock for the EIC is stopped.
1	The APBA clock for the EIC is enabled.

Bit 8 – RTC RTC APBA Clock Enable

Value	Description
0	The APBA clock for the RTC is stopped.
1	The APBA clock for the RTC is enabled.

Bit 7 – WDT WDT APBA Clock Enable

Value	Description
0	The APBA clock for the WDT is stopped.
1	The APBA clock for the WDT is enabled.

Bit 6 – GCLK GCLK APBA Clock Enable

Value	Description
0	The APBA clock for the GCLK is stopped.
1	The APBA clock for the GCLK is enabled.

Bit 5 – SUPC SUPC APBA Clock Enable

Value	Description
0	The APBA clock for the SUPC is stopped.
1	The APBA clock for the SUPC is enabled.

Bit 4 – OSC32KCTRL OSC32KCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSC32KCTRL is stopped.
1	The APBA clock for the OSC32KCTRL is enabled.

Bit 3 – OSCCTRL OSCCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSCCTRL is stopped.
1	The APBA clock for the OSCCTRL is enabled.

Bit 2 – RSTC RSTC APBA Clock Enable

Value	Description
0	The APBA clock for the RSTC is stopped.
1	The APBA clock for the RSTC is enabled.

Bit 1 – MCLK MCLK APBA Clock Enable

Value	Description
0	The APBA clock for the MCLK is stopped.
1	The APBA clock for the MCLK is enabled.

Bit 0 – PM PM APBA Clock Enable

Value	Description
0	The APBA clock for the PM is stopped.
1	The APBA clock for the PM is enabled.

13.6.8.10 APBB Mask

Name: APBBMASK
Offset: 0x18
Reset: 0x00000017
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[28:21]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[20:13]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[12:5]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Reserved[4:0]					NVMCTRL	DSU	USB
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	1	0	1	1	1

Bits 31:3 – Reserved[28:0] Reserved bits
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 2 – NVMCTRL NVMCTRL APBB Clock Enable

Value	Description
0	The APBB clock for the NVMCTRL is stopped
1	The APBB clock for the NVMCTRL is enabled

Bit 1 – DSU DSU APBB Clock Enable

Value	Description
0	The APBB clock for the DSU is stopped
1	The APBB clock for the DSU is enabled

Bit 0 – USB USB APBB Clock Enable

Value	Description
0	The APBB clock for the USB is stopped
1	The APBB clock for the USB is enabled

13.6.8.11 APBC Mask

Name: APBCMASK
Offset: 0x1C
Reset: 0x0000 7FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RFCTRL						TC1	TC0
Access	R						R	R
Reset	0						1	1

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 15 – RFCTRL RFCTRL APBC Mask Clock Enable

Value	Description
0	The APBC clock for the RFCTRL is stopped.
1	The APBC clock for the RFCTRL is enabled.

Bit 9 – TC1 TC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC1 is stopped.
1	The APBC clock for the TC1 is enabled.

Bit 8 – TC0 TC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC0 is stopped.
1	The APBC clock for the TC0 is enabled.

Bit 7 – TCC2 TCC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC2 is stopped.
1	The APBC clock for the TCC2 is enabled.

Bit 6 – TCC1 TCC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC1 is stopped.
1	The APBC clock for the TCC1 is enabled.

Bit 5 – TCC0 TCC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC0 is stopped.
1	The APBC clock for the TCC0 is enabled.

Bit 4 – SERCOM4 SERCOM4 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM4 is stopped.
1	The APBC clock for the SERCOM4 is enabled.

Bit 3 – SERCOM3 SERCOM3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM3 is stopped.
1	The APBC clock for the SERCOM3 is enabled.

Bit 2 – SERCOM2 SERCOM2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM2 is stopped.
1	The APBC clock for the SERCOM2 is enabled.

Bit 1 – SERCOM1 SERCOM1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM1 is stopped.
1	The APBC clock for the SERCOM1 is enabled.

Bit 0 – SERCOM0 SERCOM0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM0 is stopped.
1	The APBC clock for the SERCOM0 is enabled.

13.6.8.12 APBD Mask

Name: APBDMASK
Offset: 0x20
Reset: 0x000000FF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL		PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R		R/W	R/W	R/W	R/W	R/W	R/W
Reset	1		1	1	1	1	1	1

Bit 7 – CCL CCL APBD Clock Enable

Value	Description
0	The APBD clock for the CCL is stopped.
1	The APBD clock for the CCL is enabled.

Bit 5 – PTC PTC APBD Clock Enable

Value	Description
0	The APBD clock for the PTC is stopped.
1	The APBD clock for the PTC is enabled.

Bit 4 – AC AC APBD Clock Enable

Value	Description
0	The APBD clock for the AC is stopped.
1	The APBD clock for the AC is enabled.

Bit 3 – ADC ADC APBD Clock Enable

Value	Description
0	The APBD clock for the ADC is stopped.
1	The APBD clock for the ADC is enabled.

Bit 2 – TC4 TC4 APBD Clock Enable

Value	Description
0	The APBD clock for the TC4 is stopped.
1	The APBD clock for the TC4 is enabled.

Bit 1 – SERCOM5 SERCOM5 APBD Clock Enable

Value	Description
0	The APBD clock for the SERCOM5 is stopped.
1	The APBD clock for the SERCOM5 is enabled.

Bit 0 – EVSYS EVSYS APBD Clock Enable

Value	Description
0	The APBD clock for the EVSYS is stopped.
1	The APBD clock for the EVSYS is enabled.

13.6.8.13 APBE Mask

Name: APBEMASK
Offset: 0x24
Reset: 0x0000 000D
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[30:23]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[22:15]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[14:7]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Reserved[6:0]							PAC
Access	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	1	1	0	1

Bits 31:1 – Reserved[30:0] For future use
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 0 – PAC PAC APBE Clock Enable

Value	Description
0	The APBE clock for the PAC is stopped.
1	The APBE clock for the PAC is enabled.

13.7 RSTC – Reset Controller

13.7.1 Overview

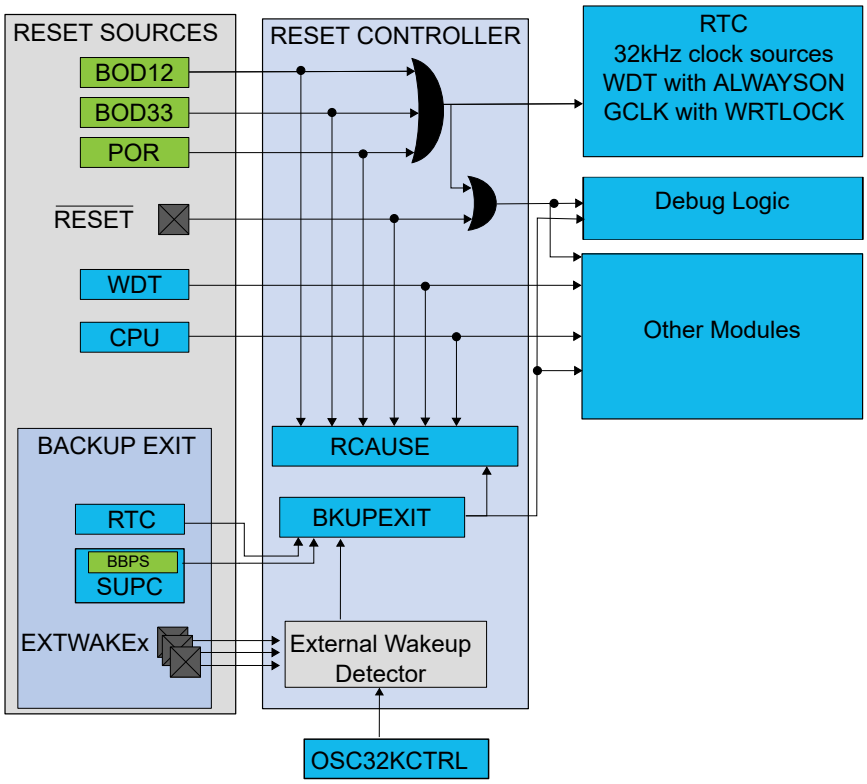
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

13.7.2 Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
  - Power supply reset sources: POR, BOD12, BOD33
  - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog reset, and System Reset Request
  - Backup exit sources: Real-Time Counter (RTC), External Wake-up (EXTWAKE), and Battery Backup Power Switch (BBPS)

13.7.3 Block Diagram

Figure 13-18. Reset System



13.7.4 Signal Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset
EXTWAKE[7:0]	Digital input	External wakeup for backup mode

One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.7.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.7.5.1 I/O Lines

Using the External Wake-up Lines requires the I/O pins to be configured in input mode before entering backup mode. External Wake-up function is active only in backup mode.

⚠ CAUTION The EXTWAKE pins can not wake up the device after it has entered Battery Backup Mode, as the I/O pin configuration is lost in this mode.

13.7.5.2 Power Management

The Reset Controller module is always on.

13.7.5.3 Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

A 32KHz clock is required to clock the RSTC if the debounce counter of the external wake-up detector is used.

Related Links

13.6 MCLK – Main Clock
13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.7.5.4 DMA

Not applicable.

13.7.5.5 Interrupts

Not applicable.

13.7.5.6 Events

Not applicable.

13.7.5.7 Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

13.7.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.7.5.9 Analog Connections

Not applicable.

13.7.6 Functional Description

13.7.6.1 Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device. External Wakeup lines causing a Backup Reset from the Backup Sleep Mode can be filtered using the debounce counter.

13.7.6.2 Basic Operation

13.7.6.2.1 Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

13.7.6.2.2 Enabling, Disabling, and Resetting

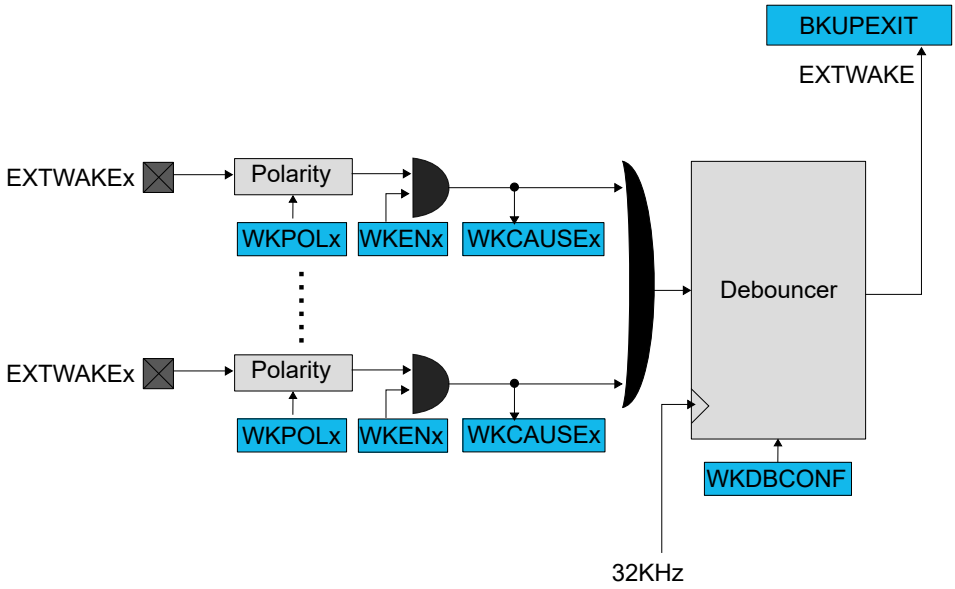
The RSTC module is always enabled.

13.7.6.2.3 External Wake-Up Detector

The External Wake-up detector is activated in Backup Sleep Mode only. In all other sleep modes, the debounce counter is stopped. Before entering Backup Mode, each external wake-up pin can be enabled by configuring the Wake-up Enable (WKEN) register. The corresponding I/O lines must also be configured in input mode using port configuration (PORT). The wake-up level can also be configured by using the Wake-up Polarity (WKPOL) register. If WKPOLx is written to 0 (default value), the input wake-up pin is active low. If WKPOLx=1 the pin is active high. All the resulting signals are wired-ORed to trigger a debounce counter which can be programmed with the Wake-up Debounce Configuration (WKDBCONF) register.

In Backup Mode, the debounce counter is running if at least one external wake-up pin is enabled and the WKDBCONF is configured to any other value than OFF. It is clocked by the OSCULP32K clock provided by the OSC32KCTRL module. If an enabled wake-up pin is asserted for a time longer than the debouncing period, the BKUPEXIT.EXTWAKE bit is set, and the value of each enable external wake-up pin is stored in the WKCAUSE register. This will allow the application to identify the external wake-up source when booting up from a backup exit reset. A backup reset is then applied. Refer to 13.7.6.2.4 Reset Causes and Effects for details.

Figure 13-19. External Wake-up Block Diagram



Related Links

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.7.6.2.4 Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence in order to determine proper action.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets
- Backup reset: Resets caused by a Backup Mode exit condition

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 13-20. Effects of the Different Reset Causes

	Power Supply Reset		User Reset		Backup Reset
	POR, BOD33	BOD12	External Reset	WDT Reset, System Reset Request	RTC, EXTWAKE, BBPS
RTC, OSC32KCTRL, RSTC, CTRLA.IORET bit of PM	Y	N	N	N	N
GCLK with WRTLOCK	Y	Y	N	N	Y
Debug logic	Y	Y	Y	N	Y
Others	Y	Y	Y	Y	Y

The external Reset is generated when pulling the $\overline{\text{RESET}}$ pin low.

The POR, BOD12, and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

From Backup Mode, the chip can be waken-up upon these conditions:

- Battery Backup Power Switch (BBPS): generated by the SUPC controller when the 3.3V VDDIO is restored.
- External wake up (EXTWAKEn): internally generated by the RSTC.
- Real-Time Counter interrupt. For details refer to the applicable INTFLAG in the RTC for details.

If one of these conditions is triggered in Backup Mode, the RCAUSE.BACKUP bit is set and the Backup Exit Register (BKUPEXIT) is updated.

Related Links

13.11 SUPC – Supply Controller

13.11.6.3 Battery Backup Power Switch

13.7.6.3 Additional Features

Not applicable.

13.7.6.4 DMA Operation

Not applicable.

13.7.6.5 Interrupts

Not applicable.

13.7.6.6 Events

Not applicable.

13.7.6.7 Sleep Mode Operation

The RSTC module is active in all sleep modes.

13.7.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RCAUSE	7:0	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
0x01	Reserved									
0x02	BKUPEXIT	7:0						BBPS	RTC	EXTWAKE
0x03	Reserved									
0x04	WKDBCONF	7:0				WKDBCNT[4:0]				
0x05 ... 0x07	Reserved									
0x08	WKPOL	7:0	WKPOL[7:0]							
		15:8								
0x0A ... 0x0B	Reserved									
0x0C	WKEN	7:0	WKEN[7:0]							
		15:8								
0x0E ... 0x0F	Reserved									
0x10	WKCAUSE	7:0	WKCAUSE[7:0]							
		15:8								

13.7.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.7.5.8 Register Access Protection.

13.7.8.1 Reset Cause

Name: RCAUSE
Offset: 0x00
Reset: Latest Reset Source
Property: –

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

Bit	7	6	5	4	3	2	1	0
	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
Access	R	R	R	R		R	R	R
Reset	x	x	x	x		x	x	x

Bit 7 – BACKUP Backup Reset
This bit is set if a Backup Reset has occurred. Refer to BKUPEXIT register to identify the source of the Backup Reset.

Bit 6 – SYST System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

Bit 5 – WDT Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

Bit 4 – EXT External Reset
This bit is set if an external Reset has occurred.

Bit 2 – BOD33 Brown Out 33 Detector Reset
This bit is set if a BOD33 Reset has occurred.

Bit 1 – BOD12 Brown Out 12 Detector Reset
This bit is set if a BOD12 Reset has occurred.

Bit 0 – POR Power On Reset
This bit is set if a POR has occurred.

13.7.8.2 Backup Exit Source

Name: BKUPEXIT
Offset: 0x02
Reset: Latest Backup Exit Source
Property: –

When a Backup Reset occurs, the bit corresponding to the exit condition is set to '1', the other bits are written to '0'.

In some specific cases, the RTC and BBPS bits can be set together, e.g. when the device leaves the battery Backup Mode caused by a BBPS condition, and a RTC event was generated during the Battery Backup Mode period.

Bit	7	6	5	4	3	2	1	0
						BBPS	RTC	EXTWAKE
Access						R	R	R
Reset						x	x	x

Bit 2 – BBPS Battery Backup Power Switch
This bit is set if the Battery Backup Power Switch of the Supply Controller changes back from battery mode to main power mode.

Bit 1 – RTC Real Timer Counter Interrupt
This bit is set if an RTC interrupt flag is set in Backup Mode.

Bit 0 – EXTWAKE External Wake-up
This bit is set if the wake-up detector has detected an external wake-up condition in Backup Mode.

Related Links

13.11 SUPC – Supply Controller
13.13 RTC – Real-Time Counter

13.7.8.3 Wakeup Debounce Configuration

Name: WKDBCONF
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WKDBCNT[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – WKDBCNT[4:0] Wakeup Debounce Counter Value
These bits define the Debounce Mode used when waking up by external wakeup pin from Backup Mode.

WKDBCNT	Name	Description
0x00	OFF	No debouncing. Input pin is low or high level sensitive depending on its WKPOLx bit.
0x01	2CK32	Input pin shall be active for at least two 32KHz clock periods.
0x02	3CK32	Input pin shall be active for at least three 32KHz clock periods.
0x03	32CK32	Input pin shall be active for at least 32 32KHz clock periods.
0x04	512CK32	Input pin shall be active for at least 512 32KHz clock periods.
0x05	4096CK32	Input pin shall be active for at least 4096 32KHz clock periods.
0x06	32768CK32	Input pin shall be active for at least 32768 32KHz clock periods.
0x07	-	Reserved

13.7.8.4 Wakeup Polarity

Name: WKPOL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKPOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – WKPOL[7:0] Wakeup Polarity
These bits define the polarity of each wakeup input pin.

Value	Description
0	Input pin x is active low.
1	Input pin x is active high.

13.7.8.5 Wakeup Enable

Name: WKEN
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection

These bits enable wakeup for input pins from Backup Mode.

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – WKEN[7:0] Wakeup Enable

Value	Description
0	The wakeup for input pin x from backup mode is disabled.
1	The wakeup for input pin x from backup mode is enabled.

13.7.8.6 Wakeup Cause

Name: WKCAUSE
Offset: 0x10
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKCAUSE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – WKCAUSE[7:0] Wakeup Cause x
This bit is updated when exiting Backup Mode.

Value	Description
0	Input pin x is not active or WKENx is written to '0'.
1	Input pin x is active and WKENx is written to '1'.

13.8 PM – Power Manager

13.8.1 Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The power domain gating technique enables the PM to turn off unused power domain supplies individually, while keeping others powered up. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption. The user can also manually control which power domains will be turned on and off in standby sleep mode.

In backup mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

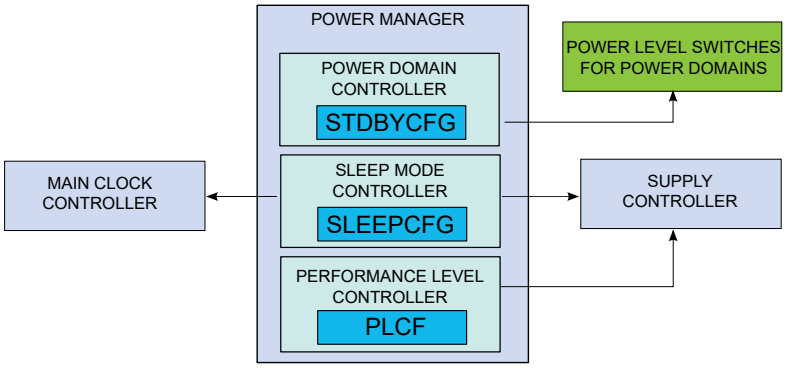
The internal state of the logic is retained (retention state) allowing the application context to be kept in non-active states.

13.8.2 Features

- Power management control
  - Sleep modes: Idle, Standby and Backup
  - Performance levels: PL0 and PL2
  - SleepWalking available in standby mode
  - Full retention state in Standby mode
  - I/O lines retention in Backup mode

13.8.3 Block Diagram

Figure 13-20. PM Block Diagram



13.8.4 Signal Description

Not applicable.

13.8.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.8.5.1 I/O Lines

Not applicable.

13.8.5.2 Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.

13.8.5.3 DMA

Not applicable.

13.8.5.4 Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.8.5.5 Events

Not applicable.

13.8.5.6 Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

If the Backup sleep mode is requested by the system while in debug mode, the core domains are kept on, and the debug modules are kept running to allow the debugger to access the internal registers. When exiting the Backup mode upon a reset condition, the core domains are reset, except the debug logic, allowing users to keep using their current debug session.

Hot plugging in standby mode is supported, except if the power domain PD2 is in retention state.

Cold or Hot plugging in Backup mode is not supported.

13.8.5.7 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag register (INTFLAG). Refer to 13.8.8.6 INTFLAG for details

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

13.8.5.8 Analog Connections

Not applicable.

13.8.6 Functional Description

13.8.6.1 Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

13.8.6.1.1 Performance Levels

To help balance between performance and power consumption, the device has two performance levels. Each of the performance levels has a maximum operating frequency and a corresponding maximum consumption in µA/MHz.

It is the application's responsibility to configure the appropriate PL depending on the application activity level. When the application selects a new PL, the voltage applied on the full logic area moves from one value to another. This voltage scaling technique allows to reduce the active power consumption while decreasing the maximum frequency of the device.

PL0

Performance Level 0 (PL0) provides the maximum energy efficiency configuration.

Refer to 15. Electrical Characteristics for details on energy consumption and maximum operating frequency.

PL2

Performance Level 2 (PL2) provides the maximum operating frequency.

Refer to 15. Electrical Characteristics for details on energy consumption and maximum operating frequency.

13.8.6.1.2 Power Domains

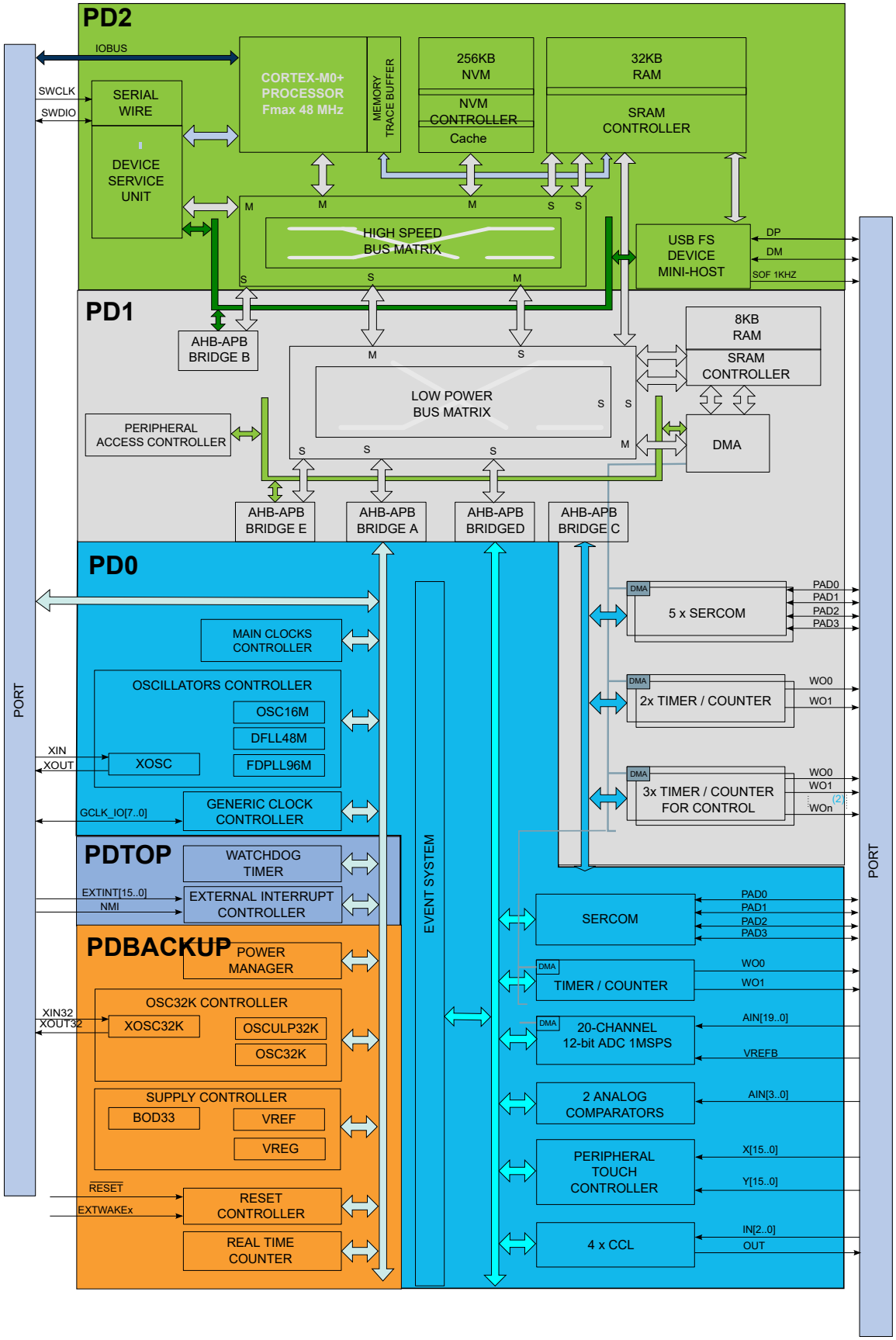
In addition to the supply domains, such as VDDIO, VDDIN and VDDANA, the device provides these power domains:

- PD0, PD1, PD2
- PDTOP
- PDBACKUP

The PD0, PD1 and PD2 are "switchable power domains". In standby or backup sleep mode, they can be turned off to save leakage consumption according to user configuration.

The three peripheral domains, PD0, PD1 and PD2, can be in retention state when none of the contained peripherals are required, but if a peripheral power domain PDn is powered, lower power domains will be powered, too. For example, if no peripherals are being used in PD2 and one or several peripherals in PD1 are active, PD2 will be powered down, PD1 will be powered and PD0 will automatically be powered, even if no peripheral is being used.

Figure 13-21. Power Domain Partitioning



PD0
PD0 is the lowest Power Domain. It contains the Event System, the Generic Clock Controller, Oscillators Controller, the Main Clocks Controller. Additionally, PD0 contains a number of peripherals that allow the device to wake up from an interrupt: one SERCOM (SERCOM5), one Timer/Counter (TC4), ADC, AC, CCL and the PTC. The PLL oscillator sources, DFLL48M and FDPLL96M, are in PD0 as well.

This power domain will automatically be activated if either PD1 or PD2 are activated.

PD1
PD1 is the intermediate Power Domain. PD1 contains the DMA controller, the Peripheral Access Controller and the Low Power Bus Matrix. It also contains the Timer/Counter for Control instances, and the low-power SRAM. PD1 contains the SERCOMs (except for SERCOM5, present in PD0) and the Timer Counters (except TC4, present in PD0).

When active, PD1 automatically activates PD0.

PD2
PD2 is the highest power domain. When activated, it will automatically activate both PD1 and PD0.

It contains the CM0+ core, the Non-Volatile Memory Controller, the Device Service Unit, USB and the SRAM.

PDTOP
PDTOP contains all controllers located in the core domain. It is powered when in Active, Idle or Standby mode. It does not have a retention mode; it is either in an active state or off. When in Backup mode, this domain is completely powered down.

PDBACKUP
The Backup Power Domain (PDBACKUP) is always on, except in the off sleep mode. It contains the 32KHz oscillator sources, the Supply Controller, the Reset Controller, the Real Time Counter, and the Power Manager itself.

13.8.6.1.3 Sleep Modes
The device can be set in a sleep mode. In sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the sleep mode depth:

- Idle sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby sleep mode: The CPU is stopped as well as the peripherals. The logic is retained, and power domain gating can be used to reduce power consumption further.
- Backup sleep mode: Only the backup domain is kept powered to allow few features to run (RTC, 32KHz clock sources, and wake-up from external pins).

13.8.6.1.4 Power Domain States and Gating
In Standby sleep mode, the Power Domain Gating technique allows for selecting the state of a PDn power domain automatically (e.g. for executing sleepwalking tasks) or manually:

Active State	The power domain is powered according to the performance level
Retention State	The main voltage supply for the power domain is switched off, while maintaining a secondary low-power supply for sequential cells. The logic context is restored when waking up.
Off State	The power domain is entirely powered off. The logic context is lost.

13.8.6.2 Principle of Operation
In active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller can save power by choosing between different sleep modes depending on application requirements, see 13.8.6.3.3 Sleep Mode Controller.

The PM Performance Level Controller allows to optimize either for low power consumption or high performance.

The PM Power Domain Controller allows for a reduction in power consumption in standby mode even further.

13.8.6.3 Basic Operation

13.8.6.3.1 Initialization
After a power-on reset, the PM is enabled, the device is in ACTIVE mode, the performance level is PL0 (the lowest power consumption) and all the power domains are in active state.

13.8.6.3.2 Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

13.8.6.3.3 Sleep Mode Controller

Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before issuing a WFI instruction.

Note: After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the INTFLAG.SLEEPRDY bit in SUPC controller is set. Before entering Standby or Backup mode, the software must ensure that the INTFLAG.SLEEPRDY bit is set.

Table 13-21. Sleep Mode Entry and Exit Table

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE _n	Synchronous(2) (APB, AHB), asynchronous (1)
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY	Synchronous(3), Asynchronous
BACKUP	SLEEPCFG.SLEEPMODE = BACKUP	Backup reset detected by the RSTC

Notes:

1. Asynchronous: Interrupt generated on generic clock, external clock or external event.
2. Synchronous: Interrupt generated on the APB clock.
3. Synchronous: Interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby and backup) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller for the power domain gating effect.

Table 13-22. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1		
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	Ultra Low- power
BACKUP	Stop	Stop	Stop	Stop	Stop	Stop	Backup regulator (ULPVREG)	OFF

Notes:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.

IDLE Mode

The IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering IDLE mode: The IDLE mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the ARM Cortex System Control register (SCR) is set, the IDLE mode will be entered

when the CPU exits the lowest priority ISR (Interrupt Service Routine, see ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the IDLE mode, the user must select the idle Sleep Mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).

- Exiting IDLE mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the ACTIVE mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and RAM are not affected by the idle sleep mode and operate in normal mode.

STANDBY Mode

The STANDBY mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in IDLE mode.
- Exiting STANDBY mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to 13.8.6.3.6 Power Domain Controller for the RAM state.

The regulator operates in low-power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

BACKUP Mode

The BACKUP mode allows achieving the lowest power consumption aside from OFF. The device is entirely powered off except for the backup domain. All peripherals in backup domain are allowed to run, e.g. the RTC can be clocked by a 32.768kHz oscillator. All PM registers are reset except the CTRLA.IORET bit.

- Entering Backup mode: This mode is entered by executing the WFI instruction after selecting the Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=BACKUP).
- Exiting Backup mode: is triggered when a Backup Reset is detected by the Reset Controller (RSTC).

13.8.6.3.4 I/O Lines Retention in BACKUP Mode

When entering BACKUP mode, the PORT is powered off but the pin configuration is retained. When the device exits the BACKUP mode, the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the CTRLA register (CTRLA.IORET).

- If IORET=0 when exiting BACKUP mode, the I/O lines configuration is released and driven by the reset value of the PORT.
- If the IORET=1 when exiting BACKUP mode, the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has programmed the PORT.

13.8.6.3.5 Performance Level

The application can change the performance level on the fly, writing to the Performance Level Select bit in the Performance Level Configuration register (PLCFG.PLSEL).

When changing to a lower performance level, the bus frequency must be reduced before writing PLCFG.PLSEL in order to avoid exceeding the limit of the target performance level.

When changing to a higher performance level, the bus frequency can be increased only after the Performance Level Ready flag in the Interrupt Flag Status and Clear (INTFLAG.PLRDY) bit set to '1', indicating that the performance level transition is complete.

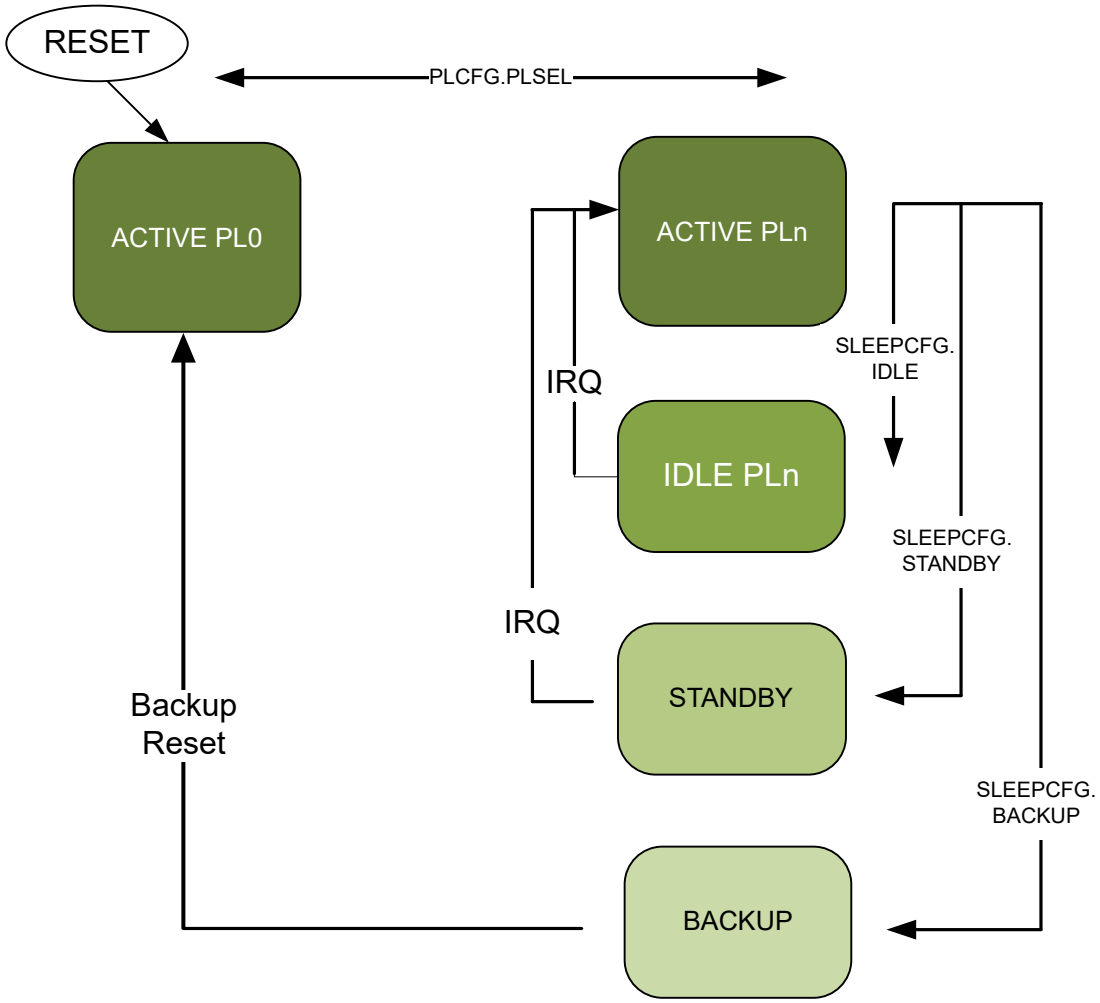
After a reset, the device starts in the lowest PL (lowest power consumption and lowest max frequency). The application can then switch to another PL at any time without any stop in the code execution. As shown in Figure 13-22, performance level transition is possible only when the device is in active mode.

The Performance Level Disable bit in the Performance Level Configuration register (PLCFG.PLDIS) can be used to freeze the performance level to PL0. This disables the performance level hardware mechanism in order to reduce both the power consumption and the wake-up start-up time from standby sleep mode.

Note: This bit, PLCFG.PLDIS, must be changed only when the current performance level is PL0.

Any attempt to modify this bit while the performance level is not PL0 is discarded, and a violation is reported to the PAC module. Any attempt to change the performance level to PLn (with n > 0) while PLCFG.PLDIS = 1 is discarded, and a violation is reported to the PAC module.

Figure 13-22. Sleep Modes and Performance Level Transitions



13.8.6.3.6 Power Domain Controller

The Power Domain Controller provides several ways of handling power domains while the device is in Standby mode or entering Standby mode:

- Default operation – All peripherals idle
  When entering Standby mode, the power domains PD0, PD1 and PD2 are set in a retention state. This allows for very low power consumption while retaining all the logic content of these power domains. When exiting Standby mode, all power domains are set back to an active state.
- Default operation – SleepWalking with static power gating (static SleepWalking)

When a peripheral needs to remain running while the device is entering standby mode (e.g., to perform a sleepwalking task, or because of its RUNSTDBY bit written to '1'), the power domain of the peripheral (PDn) remains in active state, as well as, the inferior power domains (PDm with m < n). This is an extension of the SleepWalking applied to the power domain. At the end of the sleepwalking task, the device can either be woken up or remain in standby mode.

- SleepWalking with dynamic power gating (dynamic SleepWalking)
  A power domain PDn that is in an active state due to static SleepWalking can wake up a superior power domain (PDm, with m < n) in order to perform a sleepwalking task. PDm is, then, automatically set to an active state. At the end of the sleepwalking task, either the device can be woken up, or PDm can be set again to a retention state.

The static and dynamic power gating features are fully transparent for the user. Which power domains are powered or not can also be configured manually; refer to 13.8.6.4.2 Linked Power Domains for details.

The table below illustrates these four cases to consider in standby mode:

1. SleepWalking is invoked on PD0, PD1 and PD2.
2. SleepWalking is invoked on PD0 and PD1, while PD2 is in a retention state.
3. SleepWalking is invoked on PD0, while PD1 and PD2 are in a retention state.
4. This is the default mode where all PDs are in retention state.

Table 13-23. Sleep Mode versus Power Domain State Overview

	Power Domain State				
Sleep Mode	**PD0**	**PD1**	**PD2**	**PDTOP**	**PDBACKUP**
Active	active	active	active	active	active
Idle	active	active	active	active	active
Standby – Case 1	active	active	active	active	active
Standby – Case 2	active	active	retention	active	active
Standby – Case 3	active	retention	retention	active	active
Standby – Case 4	retention	retention	retention	active	active
Backup	off	off	off	off	active

13.8.6.3.7 Regulators, RAMs and NVM State in Sleep Mode

By default, in Standby Sleep mode and backup sleep mode, the RAMs, NVM and regulators are automatically set in Low-Power mode to reduce power consumption:

- The RAM is in Low-Power mode if its power domain is in retention or off state. Refer to *RAM Automatic Low Power Mode* for details.
- Non-Volatile Memory – The NVM is located in the power domain PD2 . By default, the NVM is automatically set in low power mode in these conditions:
  - When the power domain PD2 is in a retention or off state.
  - When the device is in Standby Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in Idle Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
- Regulators: By default, in Standby Sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE.

GCLK clocks, regulators and RAM are not affected in the Idle Sleep mode and will operate as normal.

Table 13-24. Regulators, RAMs and NVM state in Sleep Mode

Sleep Mode	Switchable Power Domains			RAMs mode[1]		NVM	Regulators		
	PD0	PD1	PD2	LP SRAM	SRAM		VDDCORE		VDDBU
							main	ulp	
Active	active	active	active	normal	normal	normal	on	on	on
Idle	active	active	active	normal	auto[2]	on	on	on	on
Standby – Case 2	active	active	retention	normal	low power	low power	auto[3]	on	on
Standby – Case 3	active	retention	retention	low power	low power	low power	auto[3]	on	on
Standby – Case 4	retention	retention	retention	low power	low power	low power	off	on	on
Backup	off	off	off	off	off	off	off	off	on

Notes:

1. RAMs mode by default: STDBYCFG.BBIAS bits are set to their default value.
2. Auto: By default, NVM is in low-power mode, if not accessed.
3. Auto: By default, the main voltage regulator is on if GCLK, APBx or AHBx clock is running during SleepWalking.
4. For a description of the cases, see 13.8.6.3.6 Power Domain Controller.

13.8.6.4 Advanced Features

13.8.6.4.1 Power Domain Configuration

When entering Standby Sleep mode, a power domain is set automatically to retention state if no activity is required in it; refer to 13.8.6.3.6 Power Domain Controller for details. This behavior can be changed by writing the Power Domain Configuration bit group in the Standby Configuration register (STDBYCFG.PDCFG). For example, all power domains can be forced to remain in an active state during the Standby Sleep mode, which will accelerate wake-up time.

13.8.6.4.2 Linked Power Domains

Power domains can be linked to each other by using the Link Power Domain bit group in the Standby Configuration register (STDBYCFG.LINKPD). When PDn (n=0,1) is active, the linked power domain(s) of higher index PDm (m>n) will be in active state even if there is no activity in PDm.

When for example a static SleepWalking task is ongoing in PD0 while the device is in standby sleep mode and PD1 is linked to PD0 (LINKPD=PD01), then PD1 and PD0 are kept in active state. If dynamic SleepWalking is configured, the power state of PD1 will follow the state of PD0.

13.8.6.4.3 RAM Automatic Low Power Mode

The RAM is by default put in low power mode (back-biased) if its power domain is in retention state and the device is in standby sleep mode.

This behavior can be changed by configuring the Back Bias bit groups in the Standby Configuration register (STDBYCFG.BBIASxx), refer to the table below for details.

Note: in standby sleep mode, the DMAC can access the LP SRAM only when the power domain PD1 is not in retention and PM.STDBYCFG.BBIASLP=0x0. The DMAC can access the SRAM in standby sleep mode only when the power domain PD2 is not in retention and PM.STDBYCFG.BBIASHS=0x0.

Table 13-25. RAM Back-Biasing Mode

STBYCDFG.BBIASxx config		RAM
0x0	Retention Back Biasing mode	RAM is back-biased if its power domain is in retention state
0x1	Standby Back Biasing mode	RAM is back-biased if the device is in standby sleep mode
0x2	Standby OFF mode	RAM is OFF if the device is in standby sleep mode
0x3	Always OFF mode	RAM is OFF if its power domain is in retention state

13.8.6.4.4 Regulator Automatic Low Power Mode

In standby mode, the PM selects either the main or the low power voltage regulator to supply the VDDCORE. If all power domains are in retention state, the low power voltage regulator is used.

If a sleepwalking task is working on either asynchronous clocks (generic clocks) or synchronous clock (APB/AHB clocks), the main voltage regulator is used. This behavior can be changed by writing the Voltage Regulator Standby Mode bits in the Standby Configuration register (STDBYCFG.VREGSMOD). Refer to the following table for details.

Table 13-26. Regulator State in Sleep Mode

Sleep Modes	STDBYCFG. VREGSMOD	SleepWalking(1)	Regulator state for VDDCORE
Active	-	-	main voltage regulator
Idle	-	-	main voltage regulator
Standby (at least one PD is active)	0x0: AUTO	NO	low power regulator
		YES	main voltage regulator
	0x1: PERFORMANCE	-	main voltage regulator
	0x2: LP(2)	-(2)	low power regulator
Standby (all PD inretention)	-	-	low power regulator

Notes:

1. SleepWalking is running on GCLK clock or synchronous clock. This is not related to OSC32K, XOSC32K or OSCULP32K clocks.
2. Must only be used when SleepWalking is running on GCLK with 32KHz source.

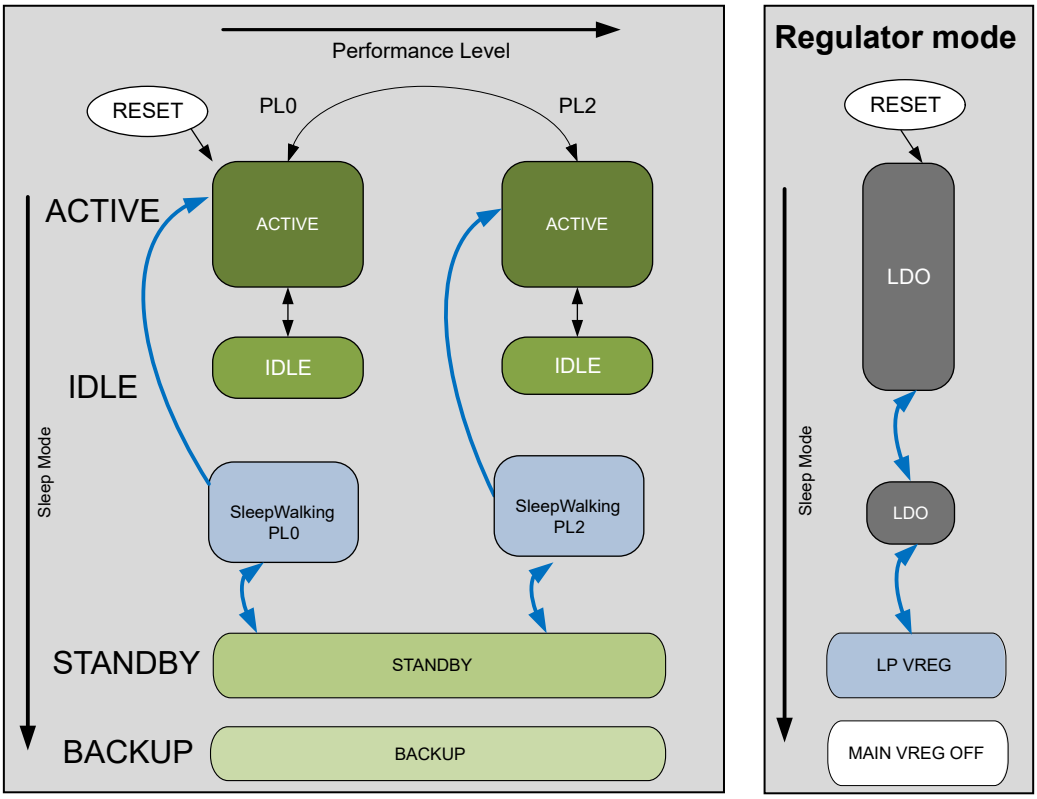
13.8.6.4.5 SleepWalking and Performance Level

SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken up by an interrupt (from a peripheral involved in SleepWalking) or enter again into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby mode, when SleepWalking is ongoing, the performance level used to execute the sleepwalking task is the current configured performance level (used in active mode).

These are illustrated in the figure below.

Figure 13-23. Operating Conditions and SleepWalking

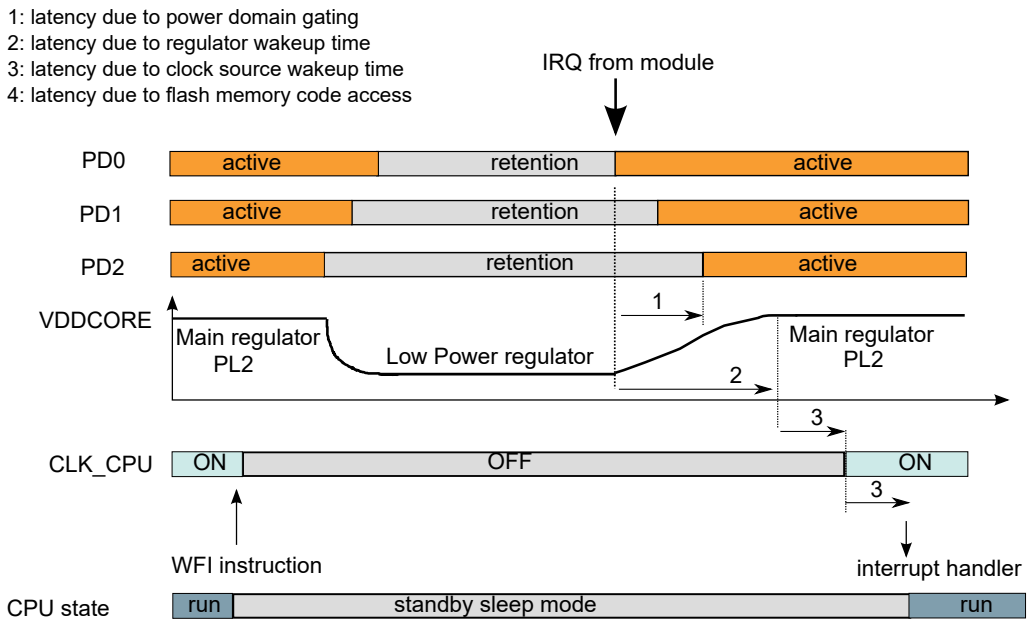


13.8.6.4.6 Wake-Up Time

As shown in the figure below, total wake-up time depends on:

- Latency due to Power Domain Gating:
  Usually, wake-up time is measured with the assumption that the power domains are already in an active state. When using Power Domain Gating, changing a power domain from a retention to active state will take a certain time; refer to Electrical Characteristics. If all power domains were already in active state in standby sleep mode, this latency is zero. If wake-up time is critical for the application, the power domains can be forced to an active state in standby sleep mode; refer to 13.8.6.4.1 Power Domain Configuration and 13.8.6.4.2 Linked Power Domains for details.
- Latency due to Performance Level and Regulator effect:
  Performance Level has to be taken into account for the global wake-up time. As an example, if PL2 is selected and the device is in standby sleep mode, the voltage level supplied by the ULP voltage regulator is lower than the one used in active mode. When the device wakes up, it takes a certain amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.
- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.

Figure 13-24. Total Wake-up Time from Standby Sleep Mode



13.8.6.5 SleepWalking with Static Power Domain Gating in Details

In standby sleep mode, the switchable power domain (PD) of a peripheral can remain in active state in order to perform sleepwalking tasks, whereas the other power domains are in retention state to reduce power consumption. This SleepWalking with static Power Domain Gating is supported by all peripherals. For some peripherals, it must be enabled by writing a Run in Standby bit in the respective Control A register (CTRLA.RUNSTDBY) to '1'. Refer to each peripheral chapter for details.

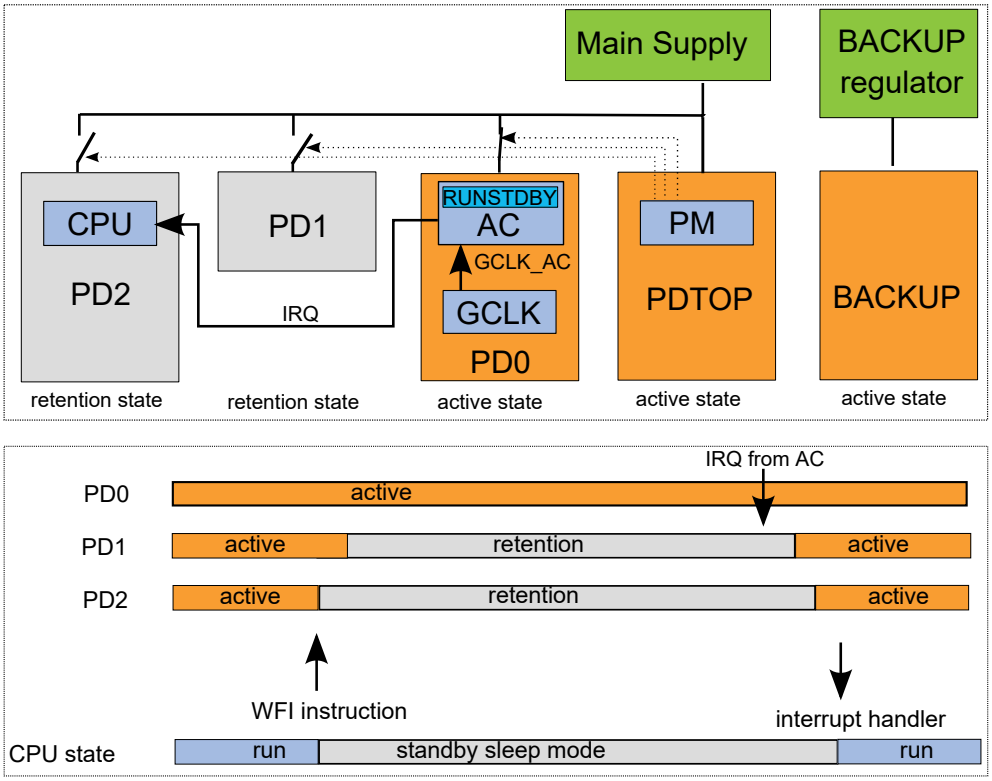
The following examples illustrate SleepWalking with static Power Domain Gating:

AC SleepWalking with Static PD Gating

The AC peripheral is used in continuous measurement mode to monitor the voltage level on input pins. An AC interrupt is generated to wake up the device. To make the AC continue to run in standby sleep mode, the RUNSTDBY bit must be written to '1'.

- Entering standby mode: As shown in the next figure, PD0 (where the AC is located) remains active, whereas PD2 and PD1 are successively set to a retention state by the Power Manager.

Figure 13-25. AC SleepWalking with Static PD Gating



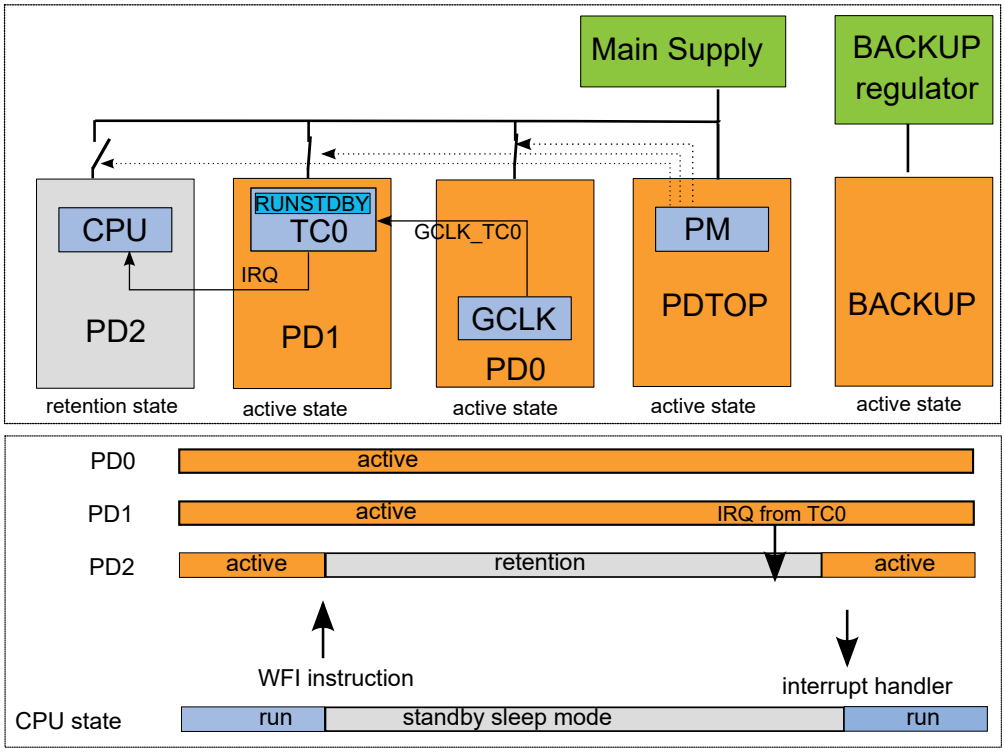
- Exiting standby mode: When conditions are met, the AC peripheral generates an interrupt to wake up the device. Successively, the PM peripheral sets PD1 and PD2 to active state. Once PD2 is in active state, the CPU is able to operate normally and execute the AC interrupt handler accordingly.
- Wake-up time:
  - The required time to set PD1 and PD2 to active state has to be considered for the global wake-up time; refer to 13.8.6.4.6 Wake-Up Time for details.
  - In this case, the VDDCORE voltage is still supplied by the main voltage regulator; refer to 13.8.6.4.4 Regulator Automatic Low Power Mode for details. Thus, global wake-up time is not affected by the regulator.

TC0 SleepWalking with Static PD Gating

TC0 peripheral is used in counter operation mode. An interrupt is generated to wake up the device based on the TC0 peripheral configuration. To make the TC0 peripheral continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'.

- Entering standby mode: As shown in Figure 13-26, PD1 (where the TC0 is located) and PD0 (where the peripheral clock generator is located) remain active, whereas PD2 is set to retention state by the Power Manager peripheral. Refer to 13.8.6.3.6 Power Domain Controller for details.
- Exiting standby mode: When conditions are met, the TC0 peripheral generates an interrupt to wake up the device. The PM peripheral sets PD2 to active state. Once PD2 is in active state, the is able to operate normally and execute the TC0 interrupt handler accordingly.
- Wake-up time:
  - The required time to set PD2 to active state has to be considered for the global wake-up time; refer to 13.8.6.4.6 Wake-Up Time for details.
  - In this case, the VDDCORE voltage is still supplied by the main voltage regulator; refer to 13.8.6.4.4 Regulator Automatic Low Power Mode for details. Thus, global wake-up time is not affected by the regulator.

Figure 13-26. TC0 SleepWalking with Static PD Gating

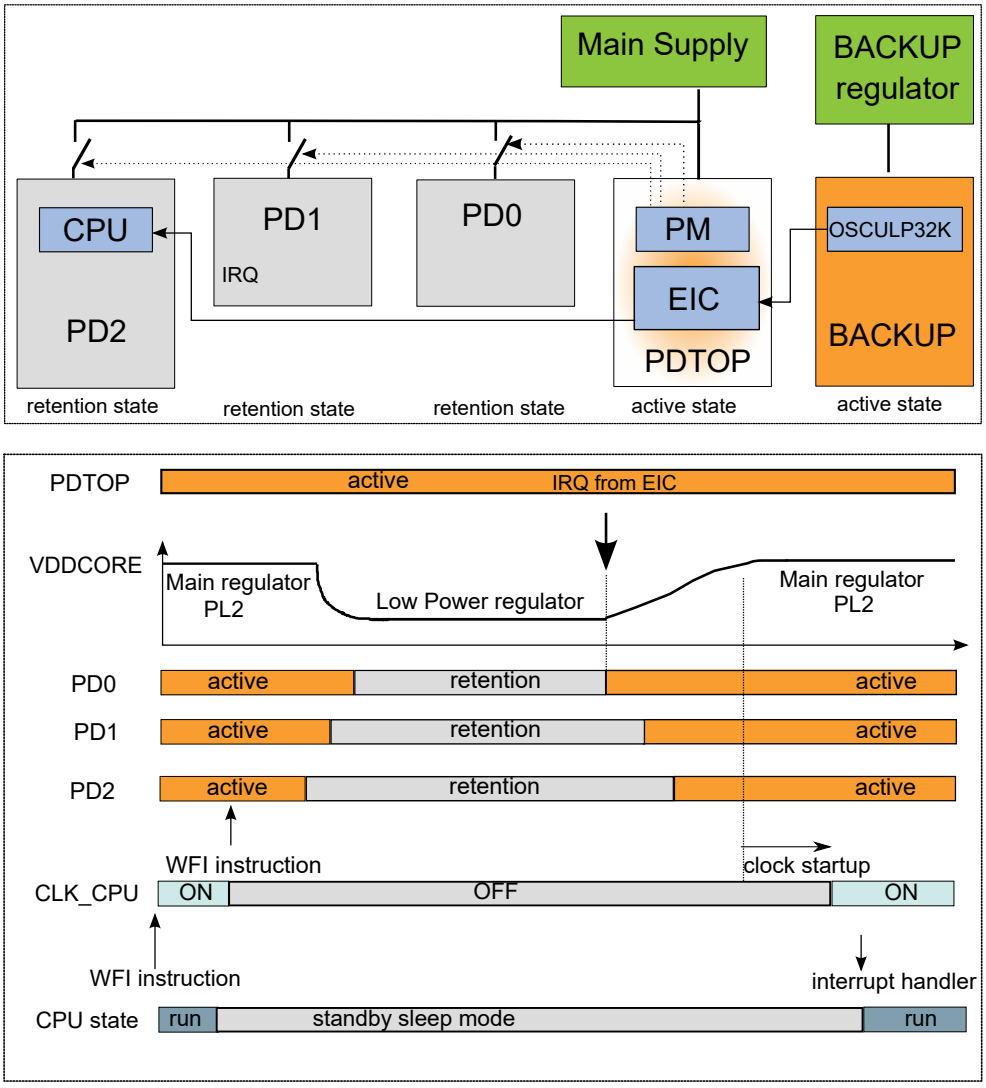


EIC SleepWalking with Static PD Gating

In this example, EIC peripheral is used to detect an edge condition to generate interrupt to the CPU. An External interrupt pin is filtered by the CLK_ULP32K clock; the GCLK peripheral is not used. Refer to 13.15 EIC – External Interrupt Controller for details. The EIC peripheral is located in the power domain PDTOP (which is not switchable), and there is no RUNSTDBY bit in the EIC peripheral.

- Entering standby mode: As shown in Figure 13-27, all the switchable power domains are set in retention state by the Power Manager peripheral. The low power regulator supplies the VDDCORE voltage level.
- Exiting standby mode: When conditions are met, the EIC peripheral generates an interrupt to wake the device up. Successively, the PM peripheral sets PD0, PD1 and PD2 to active state, and the main voltage regulator restarts. Once PD2 is in active state and the main voltage regulator is ready, the CPU is able to operate normally and execute the EIC interrupt handler accordingly.
- Wake-up time:
  - The required time to set the switchable power domains to active state has to be considered for the global wake-up time; refer to 13.8.6.4.6 Wake-Up Time for details.
  - When in standby sleep mode, the GCLK peripheral is not used, allowing the VDDCORE to be supplied by the low power regulator to reduce consumption; see 13.8.6.4.4 Regulator Automatic Low Power Mode. Consequently, the main voltage regulator wake-up time has to be considered for the global wake-up time as shown in Figure 13-27.

Figure 13-27. EIC SleepWalking with Static PD Gating



13.8.6.6 Sleepwalking with Dynamic Power Domain Gating in Details

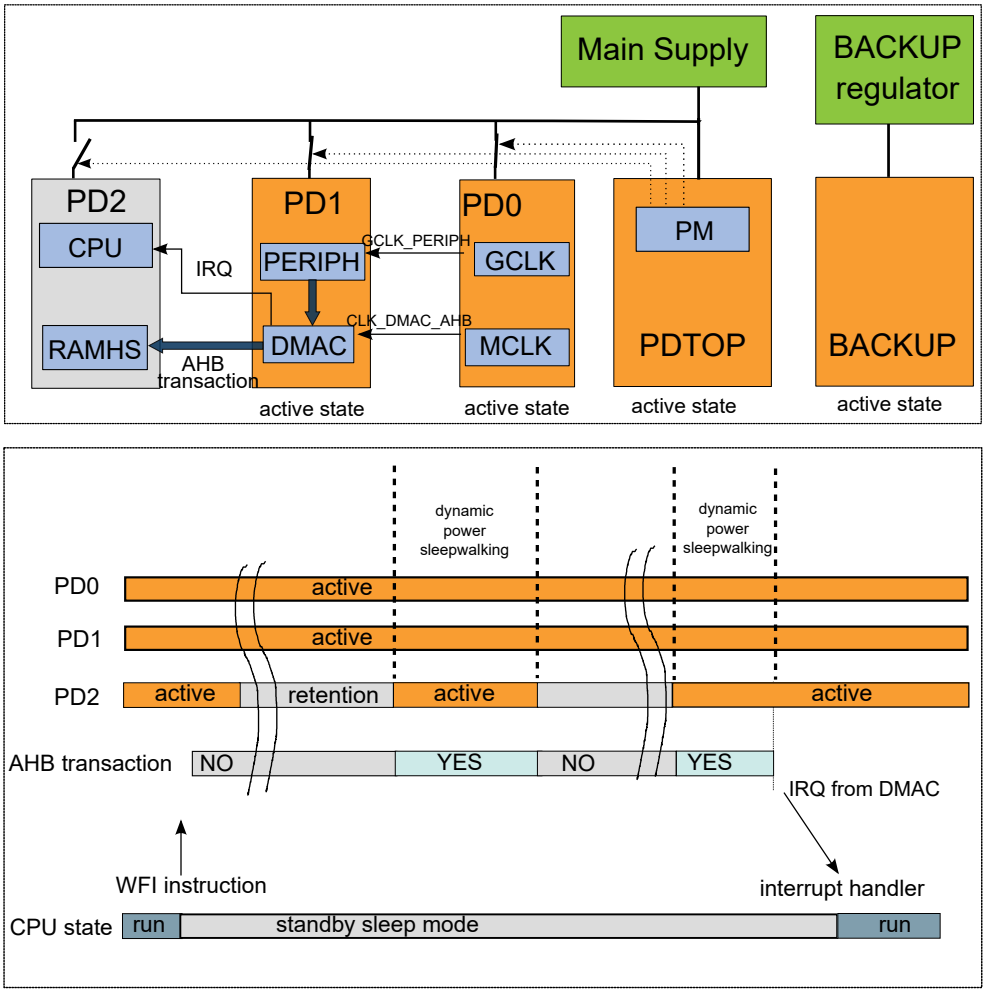
To reduce power consumption even further, Sleepwalking with dynamic Power Domain Gating (also referred to as "Dynamic Sleepwalking") is used to turn power domain state from retention to active and vice-versa, based on event or AHB bus transaction.

13.8.6.6.1 Dynamic SleepWalking on Bus Transaction

When in retention state, a power domain can be automatically set to active state by the PM if AHB bus transaction in direction to this power domain is detected. In this device, it concerns the AHB bus transaction from the DMAC to the modules located in power domain PD2.

Dynamic SleepWalking based on bus transaction is illustrated in the example below. By using the Run in Standby bit, the DMAC is configured to operate in standby sleep mode. A DMAC channel is configured to make peripheral-to-memory transfer from a module located in PD1 to the SRAM. Transfer request is triggered by the peripheral at periodic time. Refer to 13.14 DMAC – Direct Memory Access Controller for details. PD2 is set to active state only when AHB transaction is required before being set to retention state again to save power. Note that during this dynamic Sleepwalking period, the CPU is still sleeping. The device can be woken up by an interrupt, for example at the end of a complete DMA block transfer.

Figure 13-28. Dynamic SleepWalking Based on Bus Transaction



13.8.6.6.2 Dynamic SleepWalking Based on Event

To enable SleepWalking with dynamic power domain gating, the Dynamic Power Gating for Power Domain 0 and 1 bits in the Standby Configuration register (STDBYCFG.DPGPD0 and STDBYCFG.DPGPD1) have to be written to '1'.

When in retention state, a power domain can be automatically set to an active state by the PM if an event is directed to this power domain. In this device, this concerns the event users located in power domains PD1 and PD0 .

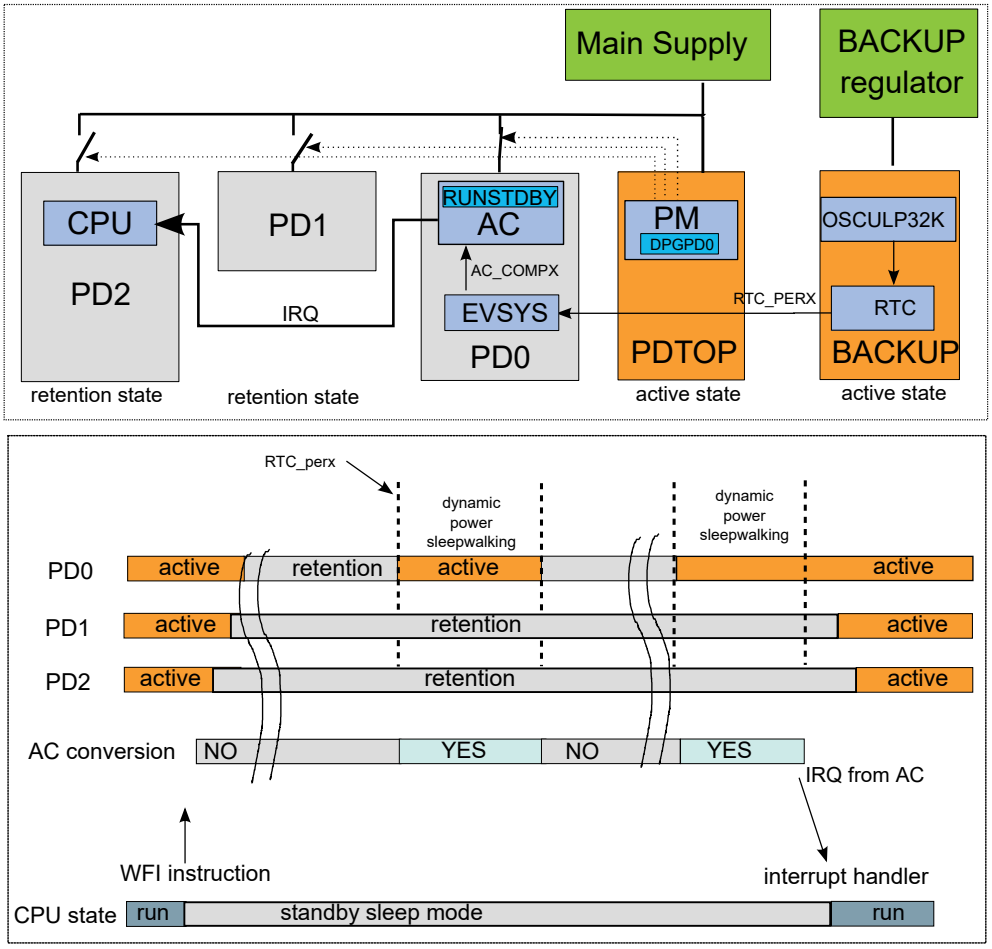
- When PD0, PD1 and PD2 are in retention state, dynamic SleepWalking can be triggered by an:
  - AC output event
  - RTC output event
  - EIC output event (if using the CLK_ULP32K clock)
- When PD0 is active, and while PD1 and PD2 are in retention state, dynamic SleepWalking can be triggered by an:
  - RTC output event
  - EIC output event (if using CLK_ULP32K)
  - All peripheral within PD0 that are capable of generating events

  All modules located in PD0 can generate events. The EVSYS event generator must be configured to either a synchronous or resynchronized path.
- When PD0 and PD1 are in retention, dynamic SleepWalking based on event is not useful.

Also refer to 13.8.6.1.2 Power Domains.

Dynamic SleepWalking based on event is illustrated in the following example:

Figure 13-29. Dynamic SleepWalking Based on Event: AC Periodic Comparison



The Analog Comparator (AC) peripheral is used in Single Shot mode to monitor voltage levels on input pins. A comparator interrupt, based on the AC peripheral configuration, is generated to wake up the device. In the GCLK module, the AC generic clock (GCLK_AC) source is routed to a 32.768 kHz oscillator (for low power applications, OSC32KULP is recommended). RTC and EVSYS modules are configured to generate periodic events to the AC. To make the comparator continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'. To enable the dynamic SleepWalking for the PD0 power domain, STDBYCFG.DPGPD0 must be written to '1'.

Entering standby mode: The Power Manager sets the PD0 power domain (where the AC module is located) in retention state, as well as PD1 and PD2. The AC comparators, COMPx, are OFF. The GCLK_AC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: The RTC event (RTC_PERX) is routed by the Event System to the Analog Comparator to trigger a single-shot measurement. This event is detected by the Power Manager, which sets the PD0 power domain to an active state, and starts the main voltage regulator.

After enabling the AC comparator and starting the GCLK_AC, the single-shot measurement can be performed during sleep mode (sleepwalking task); refer to 13.28.6.14.2 Single-Shot Measurement during Sleep for details. At the end of the conversion, if conditions to generate an interrupt are not met, the GCLK_AC clock and the AC comparator are stopped again.

The low power regulator starts again, and the PD0 power domain is set back to a retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting standby mode: During the dynamic SleepWalking sequence, if conditions are met, the AC module generates an interrupt to wake up the device. Successively, the PD1 and PD2 power domain are set to an active state by the PM.

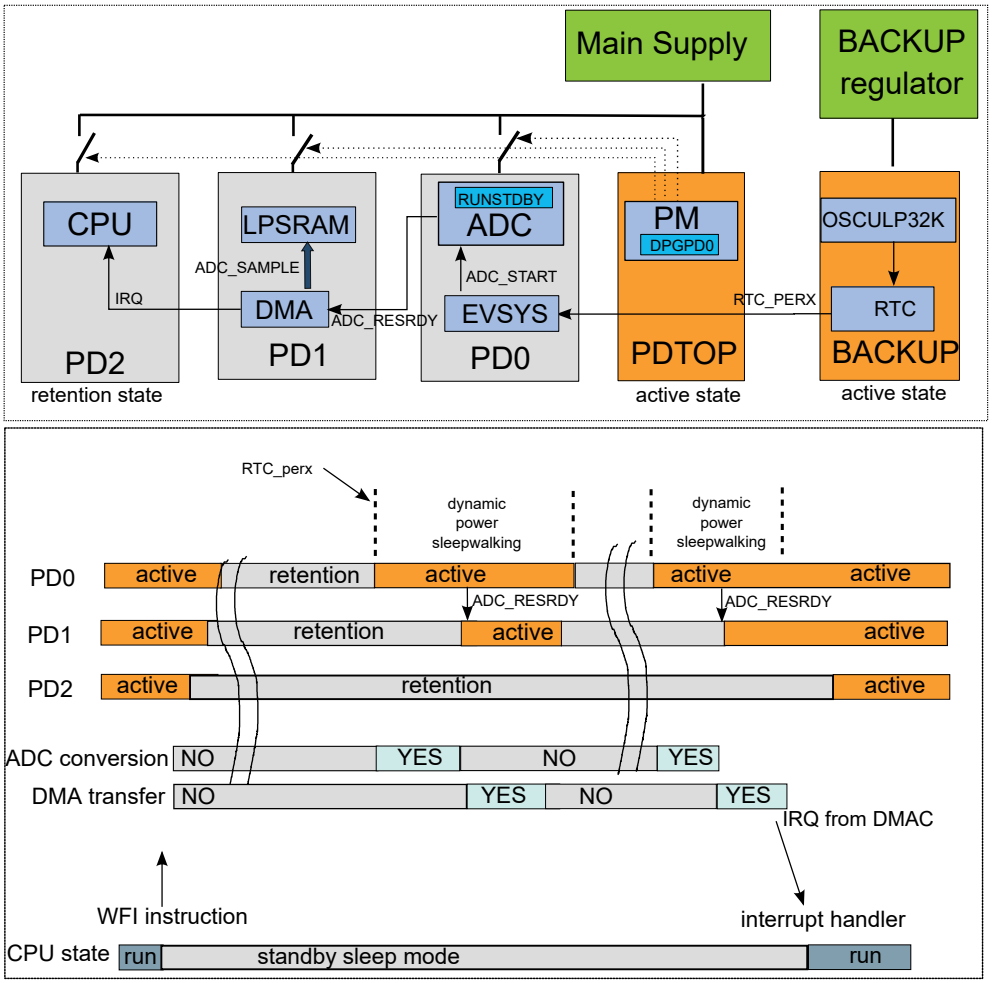
13.8.6.6.3 Dynamic SleepWalking Based on Peripheral DMA Trigger

To enable this advanced feature, the Dynamic Power Gating for Power Domain 0 and 1 bits in the Standby Configuration register (STDBYCFG.DPGPD0 and STDBYCFG.DPGPD1) have to be written to '1'.

When in retention state, the power domain PD1 (containing the DMAC) can be automatically set to active state if the PM detects a valid DMA trigger that is coming from a peripheral located in PD0. A peripheral DMA trigger is valid if the corresponding DMA channel is enabled and its Run in Standby bit (RUNSTDBY) is written to '1'.

This is illustrated in the following example:

Figure 13-30. Dynamic Sleepwalking based on Peripheral DMA Trigger



The Analog to Digital Converter (ADC) peripheral is used in one shot measurement mode to periodically convert a voltage level on input pins, and move the conversion result to RAM by DMA. After N conversions, an interrupt is generated by the DMA to wake up the device. In the GCLK module, the ADC generic clock (GCLK_ADC) source is routed to OSCULP32K. RTC and EVSYS modules are configured to generate periodic events to the ADC.

To make the ADC continue to run in standby sleep mode, its Run in Standby (RUNSTDBY) bit is written to '1'. The DMAC is configured to operate in standby sleep mode as well by using its respective RUNSTDBY bit. A DMAC channel is configured to enable peripheral-to-memory transfer from the ADC to the LPSRAM and to generate an interrupt when the block transfer is completed (after N beat transfers). The Run in Standby bit of this DMAC channel is written to '1' to allow it running in standby sleep mode.

Entering Standby mode: The Power Manager peripheral sets PD0 (where the ADC peripheral is located), PD1 (the DMAC is located here) and PD2 (CPU) to retention state. The ADC channels are OFF. The GCLK_ADC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: based on RTC conditions, a RTC event (RTC_PERX) is routed by the Event System to the ADC controller to trigger a single-shot measurement.

This event is detected by the Power Manager which sets the PD0 power domain to active state and starts the main voltage regulator.

After enabling the ADC and starting the GCLK_ADC clock, the single-shot measurement during sleep mode can be performed as a sleepwalking task, refer to the ADC documentation for details. At the end of the comparison, a DMA transfer request (ADC_RESRDY) is triggered by the ADC.

This DMA transfer request is detected by the PM, which sets PD1 (containing the DMAC) to active state. The DMAC requests the CLK_DMAC_AHB clock and transfers the sample to the memory. When the DMA beat transfer is completed, the GCLK_ADC clock and the CLK_DMAC_AHB clock are stopped again, as well as the ADC peripheral.

The low power regulator starts again and the PD0 power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting Standby mode: during SleepWalking with Dynamic Power Gating sequence, if conditions are met, the ADC peripheral generates an interrupt to wake up the device. Successively, the PD1 and PD2 power domain are set to active state by the PM.

Note: If the event trigger coming from PD0 is waking a peripheral in PD1 that does support SleepWalking, the PD1 will stay active until the follow-up task is finished. If the peripheral in PD1 does not support SleepWalking, the peripheral and PD1 will stay active after the task is finished.

Related Links

13.13 RTC – Real-Time Counter
13.18 EVSYS – Event System

13.8.6.7 DMA Operation

Not applicable.

13.8.6.8 Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 13-21 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to the Nested Vector Interrupt Controller (NVIC) for details. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

13.8.6.9 Events

Not applicable.

13.8.6.10 Sleep Mode Operation

The Power Manager is always active.

13.8.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						IORET		
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	PLCFG	7:0	PLDIS						PLSEL[1:0]	
0x03	Reserved									
0x04	INTENCLR	7:0								PLRDY
0x05	INTENSET	7:0								PLRDY
0x06	INTFLAG	7:0								PLRDY
0x07	Reserved									
0x08	STDBYCFG	7:0	VREGSMOD[1:0]		DPGPD1	DPGPD0			PDCFG[1:0]	
		15:8			BBIASLP[1:0]		BBIASHS[1:0]		LINKPD[1:0]	
0x0A ... 0x0B	Reserved									
0x0C	PWSAKDLY	7:0	IGNACK	DLYVAL[6:0]						

13.8.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.8.5.7 Register Access Protection.

13.8.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						IORET		
Access						R/W		
Reset						0		

Bit 2 – IORET I/O Retention
Note: This bit is not reset by a backup reset.

Value	Description
0	After waking up from Backup mode, I/O lines are not held.
1	After waking up from Backup mode, I/O lines are held until IORET is written to 0.

13.8.8.2 Sleep Configuration

Name: SLEEPCFG
Offset: 0x01
Reset: 0x2
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SLEEPMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SLEEPMODE[2:0] Sleep Mode
Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

Value	Name	Definition
0x0	Reserved	Reserved
0x1	Reserved	Reserved
0x2	IDLE	CPU, AHBx, and APBx clocks are OFF
0x3	Reserved	Reserved
0x4	STANDBY	ALL clocks are OFF, unless requested by sleepwalking peripheral
0x5	BACKUP	Only Backup domain is powered ON
0x6	Reserved	Reserved
0x7	Reserved	Reserved

13.8.8.3 Performance Level Configuration

Name: PLCFG
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PLDIS						PLSEL[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – PLDIS Performance Level Disable
Disabling the automatic PL selection forces the device to run in PL0 , reducing the power consumption and the wake-up time from standby sleep mode.
Changing this bit when the current performance level is not PL0 is discarded and a violation is reported to the PAC module.

Value	Description
0	The Performance Level mechanism is enabled.
1	The Performance Level mechanism is disabled.

Bits 1:0 – PLSEL[1:0] Performance Level Select

Value	Name	Definition
0x0	PL0	Performance Level 0
0x1	Reserved	Reserved
0x2	PL2	Performance Level 2
0x3	Reserved	Reserved

13.8.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Performance Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled and will generate an interrupt request when the Performance Ready Interrupt Flag is set.

13.8.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Performance Ready Interrupt Enable bit and enable the Performance Ready interrupt.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled.

13.8.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY Performance Level Ready
This flag is set when the performance level is ready and will generate an interrupt if INTENCLR/SET.PLRDY is '1'.
Writing a '1' to this bit has no effect.
Writing a '1' to this bit clears the Performance Ready interrupt flag.

13.8.8.7 Standby Configuration

Name: STDBYCFG
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			BBIASLP[1:0]		BBIASHS[1:0]		LINKPD[1:0]	
Access			R/W	R/W	R	R	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VREGSMOD[1:0]		DPGPD1	DPGPD0			PDCFG[1:0]	
Access	R	R	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bits 13:12 – BBIASLP[1:0] Back Bias for HMCRAMCLP
Refer to Table 13-25 for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode
2	Standby OFF mode
3	Always OFF mode

Bits 11:10 – BBIASHS[1:0] Back Bias for HMCRAMCHS
Refer to Table 13-25 for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode
2	Standby OFF mode
3	Always OFF mode

Bits 9:8 – LINKPD[1:0] Linked Power Domain
Refer to 13.8.6.4.2 Linked Power Domains for details.

Value	Name	Description
0x0	DEFAULT	Power domains PD0/PD1/PD2 are not linked.
0x1	PD01	Power domains PD0 and PD1 are linked. If PD0 is active, then PD1 is active even if there is no activity in PD1 .
0x2	PD12	Power domains PD1 and PD2 are linked. If PD1 is active, then PD2 is active even if there is no activity in PD2.
0x3	PD012	All Power domains are linked. If PD0 is active, then PD1 and PD2 are active even if there is no activity in PD1 or PD2.

Bits 7:6 – VREGSMOD[1:0] VREG Switching Mode
Refer to 13.8.6.4.4 Regulator Automatic Low Power Mode for details.

Value	Name	Description
0x0	AUTO	Automatic Mode
0x1	PERFORMANCE	Performance oriented
0x2	LP	Low Power consumption oriented

Bit 5 – DPGPD1 Dynamic Power Gating for Power Domain 1

Value	Description
0	Dynamic SleepWalking for power domain 1 is disabled.
1	Dynamic SleepWalking for power domain 1 is enabled.

Bit 4 – DPGPD0 Dynamic Power Gating for Power Domain 0

Value	Description
0	Dynamic SleepWalking for power domain 0 is disabled.
1	Dynamic SleepWalking for power domain 0 is enabled.

Bits 1:0 – PDCFG[1:0] Power Domain Configuration

Value	Name	Description
0x0	DEFAULT	In standby mode, all power domain switching are handled by hardware.
0x1	PD0	In standby mode, power domain 0 (PD0) is forced ACTIVE. Other power domain switching is handled by hardware.
0x2	PD01	In standby mode, power domains PD0 and PD1 are forced ACTIVE. Power domain 2 switching is handled by hardware.
0x3	PD012	In standby mode, all power domains are forced ACTIVE.

13.8.8.8 Power Switch Acknowledge Delay

Name: PWSAKDLY
Offset: 0xC
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	IGNACK	DLYVAL[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – IGNACK Ignore Acknowledge signal

Value	Description
0	Power Switch acknowledge signal is taken into account when exiting retention mode. According to the DLYVAL field, a supplementary delay is also added (from 0 to 127 digital ring oscillator period).
1	Power Switch acknowledge signal is ignored when exiting retention mode, and is replaced by an overflow counter signal clocked on internal digital ring oscillator. The overflow counter is programmable by using the DLYVAL field.

Bits 6:0 – DLYVAL[6:0] Delay value
Value of the counter overflow. See the IGNACK bit description for more details.

13.9 OSCCTRL – Oscillators Controller

13.9.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC, OSC16M, DFLL48M, and FDPLL96M.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the OSCCTRL sub-peripherals.

All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

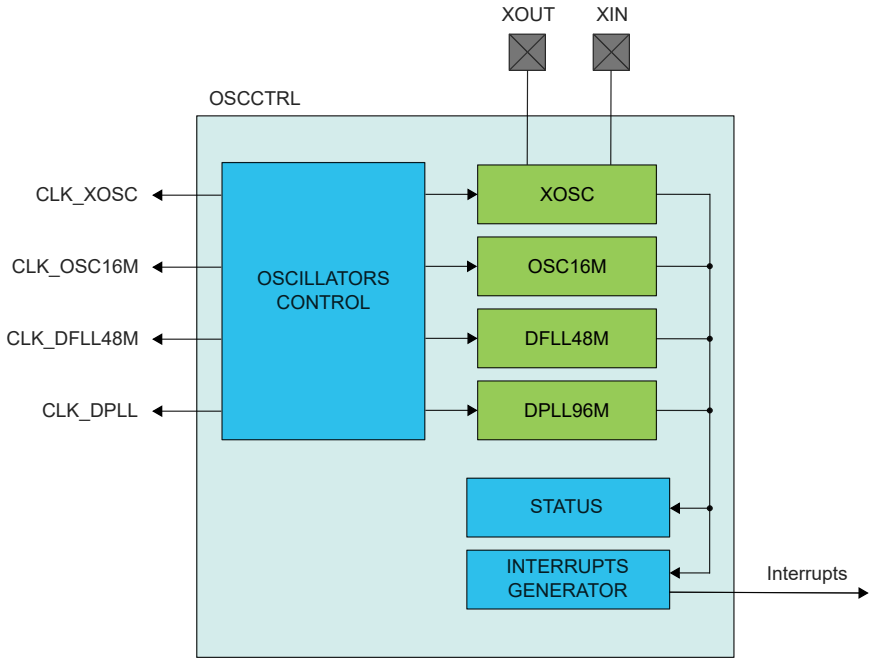
13.9.2 Features

- 0.4-32MHz Crystal Oscillator (XOSC)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
- 16MHz Internal Oscillator (OSC16M)
  - Fast startup
  - 4/8/12/16MHz output frequencies available
- Digital Frequency Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in open loop mode
  - Operates as an accurate frequency multiplier against a known frequency in closed loop mode
- Fractional Digital Phase Locked Loop (FDPLL96M)
  - 48MHz to 96MHz output frequency
  - 32kHz to 2MHz reference clock
  - A selection of sources for the reference clock
  - Adjustable proportional integral controller

– Fractional part used to achieve 1/16th of reference clock step

13.9.3 Block Diagram

Figure 13-31. OSCCTRL Block Diagram



13.9.4 Signal Description

Signal	Description	Type
XIN	Multipurpose Crystal Oscillator or external clock generator input	Analog input
XOUT	Multipurpose Crystal Oscillator output	Analog output

The I/O lines are automatically selected when XOSC is enabled.

13.9.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.9.5.1 I/O Lines

I/O lines are configured by OSCCTRL when XOSC is enabled, and need no user configuration.

13.9.5.2 Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.9.5.3 Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSC, OSC16M, DFLL48M, and FDPLL96M.

The OSCCTRL bus clock (CLK_OSCCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

The DFLL48M control logic uses the DFLL oscillator output, which is also asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to 13.9.6.9 Synchronization for further details.

Related Links
13.6 MCLK – Main Clock

13.9.5.4 DMA
Not applicable.

13.9.5.5 Interrupts
The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

Related Links
11.2 Nested Vector Interrupt Controller

13.9.5.6 Debug Operation
When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.9.5.7 Register Access Protection
All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.9.5.8 Analog Connections
The 0.4-32MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

13.9.6 Functional Description

13.9.6.1 Principle of Operation
XOSC, OSC16M, DFLL48M, and FDPLL96M are configured via OSCCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The Status register gathers different status signals coming from the sub-peripherals controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

13.9.6.2 External Multipurpose Crystal Oscillator (XOSC) Operation
The XOSC can operate in two different modes:

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 0.4-32MHz crystal

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pin will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.ENABLE).

To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRL.XTALEN) must written to '1'. If XOSCCTRL.XTALEN is zero, the external clock input on XIN will be enabled.

When in crystal oscillator mode (XOSCCTRL.XTALEN=1), the External Multipurpose Crystal Oscillator Gain (XOSCCTRL.GAIN) must be set to match the external crystal oscillator frequency. If the External Multipurpose

Crystal Oscillator Automatic Amplitude Gain Control (XOSCCTRL.AMPGC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in a lower power consumption.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRL.RUNSTDBY, XOSCCTRL.ONDEMAND, and XOSCCTRL.ENABLE. If XOSCCTRL.ENABLE=0, the XOSC will be always stopped. For XOSCCTRL.ENABLE=1, this table is valid:

Table 13-27. XOSC Sleep Behavior

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

After a hard reset, or when waking up from a sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRL.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set once the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Related Links

13.5 GCLK - Generic Clock Controller

13.9.6.3 16MHz Internal Oscillator (OSC16M) Operation

The OSC16M is an internal oscillator operating in open-loop mode and generating 4, 8, 12, or 16MHz frequency. The OSC16M frequency is selected by writing to the Frequency Select field in the OSC16M register (OSC16MCTRL.FSEL). OSC16M is enabled by writing '1' to the Oscillator Enable bit in the OSC16M Control register (OSC16MCTRL.ENABLE), and disabled by writing a '0' to this bit. Frequency selection must be done when OSC16M is disabled.

After enabling OSC16M, the OSC16M clock is output as soon as the oscillator is ready (STATUS.OSC16MRDY=1). User must ensure that the OSC16M is fully disabled before enabling it by reading STATUS.OSC16MRDY=0.

After reset, OSC16M is enabled and serves as the default clock source at 4MHz.

OSC16M will behave differently in different sleep modes based on the settings of OSC16MCTRL.RUNSTDBY, OSC16MCTRL.ONDEMAND, and OSC16MCTRL.ENABLE. If OSC16MCTRL.ENABLE=0, the OSC16M will be always stopped. For OSC16MCTRL.ENABLE=1, this table is valid:

Table 13-28. OSC16M Sleep Behavior

CPU Mode	OSC16MCTRL.RUNSTDBY	OSC16MCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

OSC16M is used as a clock source for the generic clock generators. This is configured by the Generic Clock Generator Controller.

Related Links

13.5 GCLK - Generic Clock Controller

13.9.6.4 Digital Frequency Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.

Related Links

13.5 GCLK - Generic Clock Controller

13.9.6.4.1 Basic Operation

Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M clock, CLK_DFLL48M, will be determined by the values written to the DFLL Coarse Value bit group and the DFLL Fine Value bit group (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register. Using "DFLL48M COARSE CAL" value from the Non Volatile Memory Software Calibration Area in DFLL.COARSE helps to output a frequency close to 48MHz.

It is possible to change the values of DFLLVAL.COARSE and DFLLVAL.FINE while the DFLL48M is enabled and in use, and thereby to adjust the output frequency of CLK_DFLL48M.

Related Links

10.3 NVM User Row Mapping

Closed-Loop Operation

In closed-loop operation, the DFLL48M output frequency is continuously regulated against a precise reference clock of relatively low frequency. This will improve the accuracy and stability of the CLK_DFLL48M clock in comparison to the open-loop (free-running) configuration.

Before closed-loop operation can be enabled, the DFLL48M must be enabled and configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference).
2. Select the maximum step size allowed for finding the Coarse and Fine values by writing the appropriate values to the DFLL Coarse Maximum Step and DFLL Fine Maximum Step bit groups (DFLLMUL.CSTEP and DFLLMUL.FSTEP) in the DFLL Multiplier register.
   A small step size will ensure low overshoot on the output frequency, but it will typically take longer until locking is achieved. A high value might give a large overshoot, but will typically provide faster locking.

   DFLLMUL.CSTEP and DFLLMUL.FSTEP should not be higher than 50% of the maximum value of DFLLVAL.COARSE and DFLLVAL.FINE, respectively.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register.
   Note: When choosing DFLLMUL.MUL, the output frequency must not exceed the maximum frequency of the device.
   If the target frequency is below the minimum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum frequency.
4. Start the closed loop mode by writing '1' to the DFLL Mode Selection bit in the DFLL Control register (DFLLCTRL.MODE). See Frequency Locking for details.

The frequency of CLK_DFLL48M ($F_{clkdfll48m}$) is given by:

$$F_{clkdfll48m} = DFLLMUL \cdot MUL \times F_{clkdfll48m\_ref}$$

where $F_{clkdfll48m\_ref}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

Related Links
13.5 GCLK - Generic Clock Controller

Frequency Locking
After enabling closed-loop operation by writing DFLLCTRL.MODE=1, the Coarse Value and the Fine Value bit fields in the DFLL48M Value register (DFLLVAL.COARSE and DFLLVAL.FINE) are used as starting parameters for the locking procedure.

Note: DFLLVAL.COARSE and DFLLVAL.FINE are read-only in closed-loop mode, and are controlled by the frequency tuner to meet user specified frequency.

The frequency locking is divided into two stages: coarse and fine lock.

Coarse Lock. Starting from the original DFLLVAL.COARSE and DFLLVAL.FINE, the control logic quickly finds the correct value for DFLLVAL.COARSE and sets the output frequency to a value close to the correct frequency. On coarse lock, the DFLL Locked on Coarse Value bit (STATUS.DFLLLCKC) in the Status register will be set.

Fine Lock. In this stage, the control logic tunes the value in DFLLVAL.FINE so that the output frequency is very close to the desired frequency. On fine lock, the DFLL Locked on Fine Value bit (STATUS.DFLLLCKF) in the Status register will be set.

Interrupts are generated by STATUS.DFLLLCKC and STATUS.DFLLLCKF, if INTENSET.DFLLLCKC or INTENSET.DFLLLCKF, respectively, are written to '1'.

The accuracy of the output frequency depends on which locks are set.

Note: Writing DFLLVAL.COARSE to a value close to the final value before entering closed-loop mode will reduce the time needed to get a lock on Coarse.

For a DFLL48M output frequency of 48MHz, the bit field "DFLL48M COARSE CAL" in the NVM Software Calibration Area provides a matching value for DFLL.COARSE, and will start DFLL with a frequency close to 48MHz.

This procedure will reduce the locking time to only the DFLL Fine Lock time:

1. Load the "DFLL48M COARSE CAL" value from the NVM Software Calibration Area into the DFLL.COARSE bit field.
2. Enable the Bypass Coarse Lock (DFLLCTRL.BPLCKC=1).
3. Start DFLL close loop (DFLLCTRL.MODE=1).

Related Links
10.3 NVM User Row Mapping

Frequency Error Measurement
The ratio between CLK_DFLL48M_REF and CLK48M_DFLL is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF) in the DFLL Value register.

The relative error of CLK_DFLL48M with respect to the target frequency is calculated as follows:

$$ERROR = \frac{\text{DFLLVAL.DIFF}}{\text{DFLLMUL.MUL}}$$

Drift Compensation
If the Stable DFLL Frequency bit (DFLLCTRL.STABLE) in the DFLL Control register is '0', the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks.

Note: This means that DFLLVAL.FINE can change after every measurement of CLK_DFLL48M.

The DFLLVAL.FINE value may overflow or underflow in closed-loop mode due to large drift/instability of the clock source reference, and the DFLL Out Of Bounds bit (STATUS.DFLLOOB) in the Status register will be set. After an Out of Bounds error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency.

A zero-to-one transition of STATUS.DFLLOOB will generate an interrupt, if the DFLL Out Of Bounds bit in the Interrupt Enable Set register (INTENSET.DFLLOOB) is '1'. This interrupt will also be set if the tuner is not able to lock on the correct Coarse value.

To avoid this out-of-bounds error, the reference clock must be stable; an external oscillator XOSC32K is recommended.

Reference Clock Stop Detection
If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2 * MUL_{MAX})), the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will be set.

Detecting a stopped reference clock can take a long time, in the order of 217 CLK_DFLL48M cycles.

When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume when the CLK_DFLL48M_REF is restarted.

A zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will generate an interrupt, if the DFLL Reference Clock Stopped bit in the Interrupt Enable Set register (INTENSET.DFLLRCS) is '1'.

13.9.6.4.2 Additional Features

Dealing with Settling Time in Closed-Loop Mode
The time from selecting a new CLK_DFLL48M output frequency until this frequency is output by the DFLL48M can be up to several microseconds. A small value in DFLLMUL.MUL can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving locks.

To avoid this, a chill cycle can be enabled, during which the CLK_DFLL48M frequency is not measured. The chill cycle is enabled by default, but can be disabled by writing '1' to the DFLL Chill Cycle Disable bit in the DFLL Control register (DFLLCTRL.CCDIS). Enabling chill cycles might double the lock time.

Another solution to this problem is using less strict lock requirements. This is called Quick Lock (QL). QL is enabled by default as well, but it can be disabled by writing '1' to the Quick Lock Disable bit in the DFLL Control register (DFLLCTRL.QLDIS). The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

USB Clock Recovery Module
USB Clock Recovery mode can be used to create the 48MHz USB clock from the USB Start Of Frame (SOF). This mode is enabled by writing a '1' to both the USB Clock Recovery Mode bit and the Mode bit in DFLL Control register (DFLLCTRL.USBCRM and DFLLCTRL.MODE).

The SOF signal from USB device will be used as reference clock (CLK_DFLL_REF), ignoring the selected generic clock reference. When the USB device is connected, a SOF will be sent every 1ms, thus DFLLVAL.MUX bits should be written to 0xBB80 to obtain a 48MHz clock.

In USB clock recovery mode, the DFLLCTRL.BPLCKC bit state is ignored, and the value stored in the DFLLVAL.COARSE will be used as final Coarse Value. The COARSE calibration value can be loaded from NVM OTP row by software. The locking procedure will also go instantaneously to the fine lock search.

The DFLLCTRL.QLDIS bit must be cleared and DFLLCTRL.CCDIS should be set to speed up the lock phase. The DFLLCTRL.STABLE bit state is ignored, an auto jitter reduction mechanism is used instead.

Wake from Sleep Modes
DFLL48M can optionally reset its lock bits when it is disabled. This is configured by the Lose Lock After Wake bit in the DFLL Control register (DFLLCTRL.LLAW).

If DFLLCTRL.LLAW is zero, the DFLL48M will be re-enabled and start running with the same configuration as before being disabled, even if the reference clock is not available. The locks will not be lost. After the reference clock has restarted, the fine lock tracking will quickly compensate for any frequency drift during sleep if DFLLCTRL.STABLE is zero.

If DFLLCTRL.LLAW is '1' when disabling the DFLL48M, the DFLL48M will lose all its locks, and needs to regain these through the full lock sequence.

Accuracy
There are three main factors that determine the accuracy of $F_{clkdfll48m}$. These can be tuned to obtain maximum accuracy when fine lock is achieved.

- Fine resolution. The frequency step between two Fine values. This is relatively smaller for higher output frequencies.

- Resolution of the measurement: If the resolution of the measured $F_{clkdfll48m}$ is low, i.e., the ratio between the CLK_DFLL48M frequency and the CLK_DFLL48M_REF frequency is small, the DFLL48M might lock at a frequency that is lower than the targeted frequency. It is recommended to use a reference clock frequency of 32KHz or lower to avoid this issue for low target frequencies.
- The accuracy of the reference clock.

13.9.6.5 Digital Phase Locked Loop (DPLL) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency, CLK_DPLL, via phase comparison. The DPLL controller supports three independent sources of reference clocks:

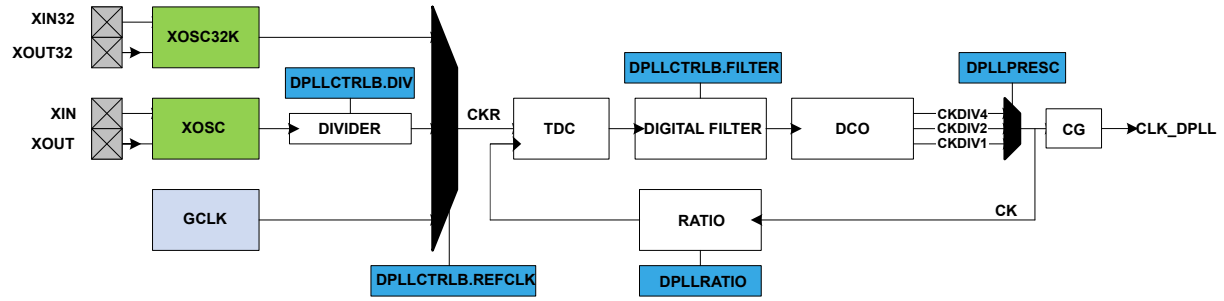
- XOSC32K: this clock is provided by the 32K External Crystal Oscillator (XOSC32K).
- XOSC: this clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK: this clock is provided by the Generic Clock Controller.

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

$$f_{CK} = f_{CKR} \times \left(\mathrm{LDR} + 1 + \frac{\mathrm{LDRFRAC}}{16}\right) \times \frac{1}{2^{\mathrm{PRESC}}}$$

Where f_{CK} is the frequency of the DPLL output clock, LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part, f_{CKR} is the frequency of the selected reference clock, and PRESC is the output prescaler value.

Figure 13-32. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. Note that the fractional part has a negative impact on the jitter of the DPLL.

> Example (integer mode only): assuming F_{CKR} = 32kHz and F_{CK} = 48MHz, the multiplication ratio is 1500. It means that LDR shall be set to 1499.

> Example (fractional mode): assuming F_{CKR} = 32kHz and F_{CK} = 48.006MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

Related Links

13.5 GCLK - Generic Clock Controller

13.10 OSC32KCTRL – 32KHz Oscillators Controller

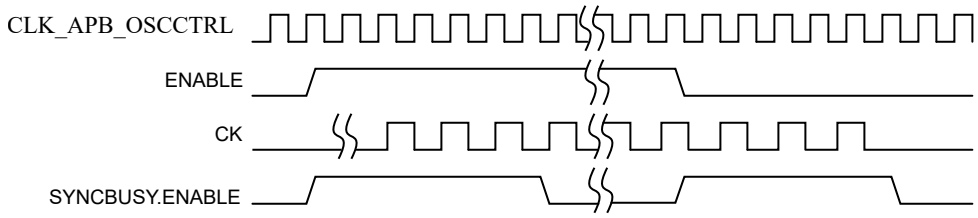
13.9.6.5.1 Basic Operation

Initialization, Enabling, Disabling, and Resetting

The DPLLC is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLLC is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 13-33. Enable Synchronization Busy Operation



The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.

When the Wake Up Fast bit (DPLLCTRLB.WUF) is set, the wake up fast mode is activated. In this mode the clock gating cell is enabled at the end of the startup time. At this time the final frequency is not stable, as it is still during the acquisition period, but it allows to save several milliseconds. After first acquisition, the Lock Bypass bit (DPLLCTRLB.LBYPASS) indicates if the lock signal is discarded from the control of the clock gater (CG) generating the output clock CLK_DPLL.

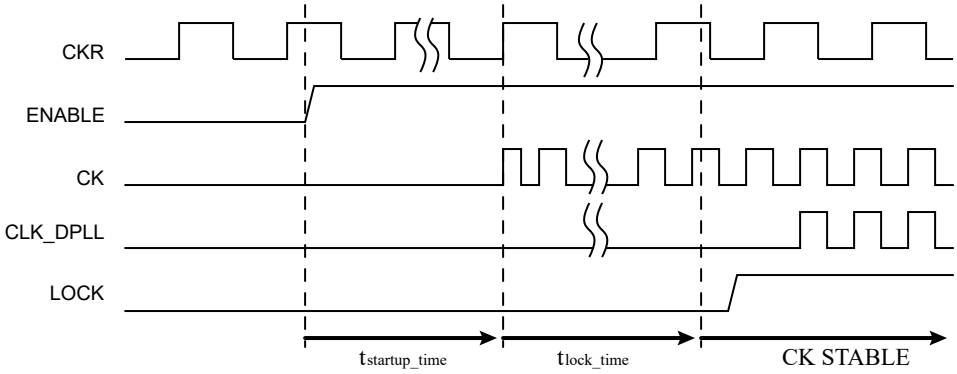
Table 13-29. CLK_DPLL Behavior from Startup to First Edge Detection

WUF	LTIME	CLK_DPLL Behavior
0	0	Normal Mode: First Edge when lock is asserted
0	Not Equal To Zero	Lock Timer Timeout mode: First Edge when the timer down-counts to 0.
1	X	Wake Up Fast Mode: First Edge when CK is active (startup time)

Table 13-30. CLK_DPLL Behavior after First Edge Detection

LBYPASS	CLK_DPLL Behavior
0	Normal Mode: the CLK_DPLL is turned off when lock signal is low.
1	Lock Bypass Mode: the CLK_DPLL is always running, lock is irrelevant.

Figure 13-34. CK and CLK_DPLL Output from DPLL Off Mode to Running Mode



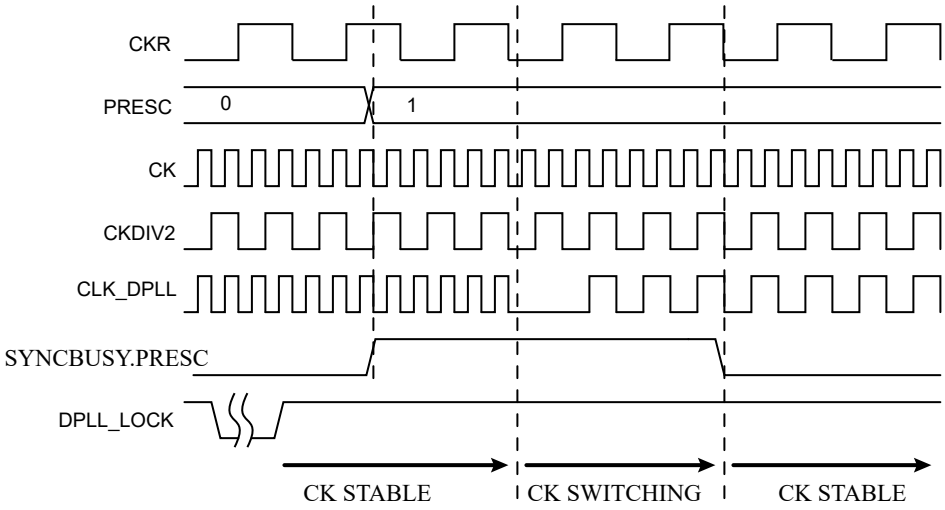
Reference Clock Switching

When a software operation requires reference clock switching, the recommended procedure is to turn the DPLL into the standby mode, modify the DPLLCTRLB.REFCLK to select the desired reference source, and activate the DPLL again.

Output Clock Prescaler

The DPLL controller includes an output prescaler. This prescaler provides three selectable output clocks CK, CKDIV2 and CKDIV4. The Prescaler bit field in the DPLL Prescaler register (DPLLPRESC.PRESC) is used to select a new output clock prescaler. When the prescaler field is modified, the DPLLSYNCBUSY.DPLLPRESC bit is set. It will be cleared by hardware when the synchronization is over.

Figure 13-35. Output Clock Switching Operation

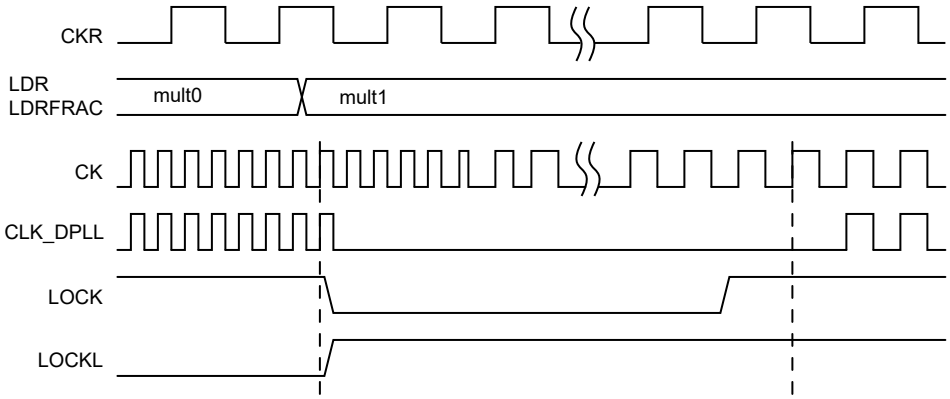


Loop Divider Ratio Updates

The DPLL Controller supports on-the-fly update of the DPLL Ratio Control (DPLLRATIO) register, allowing to modify the loop divider ratio and the loop divider ratio fractional part when the DPLL is enabled.

STATUS.DPLLLDRTO is set when the DPLLRATIO register has been modified and the DPLL analog cell has successfully sampled the updated value. At that time the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

Figure 13-36. RATIOCTRL register update operation



Digital Filter Selection

The PLL digital filter (PI controller) is automatically adjusted in order to provide a good compromise between stability and jitter. Nevertheless a software operation can override the filter setting using the Filter bit field in the DPLL Control B register (DPLLCTRLB.FILTER). The Low Power Enable bit (DPLLCTRLB.LPEN) can be use to bypass the Time to Digital Converter (TDC) module.

13.9.6.6 DMA Operation

Not applicable.

13.9.6.7 Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDY - Multipurpose Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSCRDY bit is detected
- CLKFAIL - Clock Failure. A 0-to-1 transition on the STATUS.CLKFAIL bit is detected
- OSC16MRDY - 16MHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC16MRDY bit is detected
- DFLL-related:
  - DFLLRDY - DFLL48M Ready: A 0-to-1 transition of the STATUS.DFLLRDY bit is detected
  - DFLLOOB - DFLL48M Out Of Boundaries: A 0-to-1 transition of the STATUS.DFLLOOB bit is detected
  - DFLLLOCKF - DFLL48M Fine Lock: A 0-to-1 transition of the STATUS.DFLLLOCKF bit is detected
  - DFLLLOCKC - DFLL48M Coarse Lock: A 0-to-1 transition of the STATUS.DFLLLOCKC bit is detected
  - DFLLRCS - DFLL48M Reference Clock has Stopped: A 0-to-1 transition of the STATUS.DFLLRCS bit is detected
- DPLL-related:
  - DPLLLOCKR - DPLL Lock Rise: A 0-to-1 transition of the STATUS.DPLLLOCKR bit is detected
  - DPLLLOCKF - DPLL Lock Fall: A 0-to-1 transition of the STATUS.DPLLLOCKF bit is detected
  - DPLLLTTO - DPLL Lock Timer Time-out: A 0-to-1 transition of the STATUS.DPLLLTTO bit is detected
  - DPLLLDRTO - DPLL Loop Divider Ratio Update Complete. A 0-to-1 transition of the STATUS.DPLLLDRTO bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: The interrupts must be globally enabled for interrupt requests to be generated.

13.9.6.8 Events

Not applicable.

13.9.6.9 Synchronization

DFLL48M

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains.

Once the DFLL is enabled, any read and write operation requires the DFLL Ready bit in the Status register (STATUS.DFLLRDY) to read '1'.

Note: Once the DFLL48M is enabled in on-demand mode (DFLLCTRL.ONDEMAND=1), the STATUS.DFLLRDY bit will keep to '0' until the DFLL48M is requested by a peripheral.

Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (STATUS.DFLLRDY) is set to '1'. When this bit is set, the DFLL48M can be configured and CLK_DFLL48M is ready to be used. Any write to any of the DFLL48M control registers while DFLLRDY is '0' will be ignored.

In order to read from the DFLLVAL register in closed loop mode, the user must request a read synchronization by writing a '1' to the Read Request bit in the DFLL Synchronization register (DFLLSYNC.READREQ). This is required because the DFLL controller may change the content of the DFLLVAL register any time. If a read operation is issued while the DFLL controller is updating the DFLLVAL content, a zero will be returned.

Note: Issuing a read on any register while a write-synchronization is still on-going will return a zero.

Read-Synchronized registers using DFLLSYNC.READREQ:

- DFLL48M Value register (DFLLVAL)

Write-Synchronized registers:

- DFLL48M Control register (DFLLCTRL)
- DFLL48M Value register (DFLLVAL)
- DFLL48M Multiplier register (DFLLMUL)

DPLL96M

Due to the multiple clock domains, some registers in the DPLL96M must be synchronized when accessed.

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLCTRLA.ENABLE)
- DPLL Ratio register (DPLLRATIO)
- DPLL Prescaler register (DPLLPRESC)

Related Links

13.4.3 Register Synchronization

13.9.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	7:0				OSC16MRDY				XOSCRDY
		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x04	INTENSET	7:0				OSC16MRDY				XOSCRDY
		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x08	INTFLAG	7:0				OSC16MRDY				XOSCRDY
		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x0C	STATUS	7:0				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		31:24								
0x10	XOSCCTRL	7:0	ONDEMAND	RUNSTDBY				XTALEN	ENABLE	
		15:8	STARTUP[3:0]				AMPGC	GAIN[2:0]		
0x12 ... 0x13	Reserved									
0x14	OSC16MCTRL	7:0	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
0x15 ... 0x17	Reserved									
0x18	DFLLCTRL	7:0	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
		15:8					WAITLOCK	BPLCKC	QLDIS	CCDIS
0x1A ... 0x1B	Reserved									
0x1C	DFLLVAL	7:0	FINE[7:0]							
		15:8	COARSE[5:0]						FINE[9:8]	
		23:16	DIFF[7:0]							
		31:24	DIFF[15:8]							
0x20	DFLLMUL	7:0	MUL[7:0]							
		15:8	MUL[15:8]							
		23:16	FSTEP[7:0]							
		31:24	CSTEP[5:0]						FSTEP[9:8]	
0x24	DFLLSYNC	7:0	READREQ							
0x25 ... 0x27	Reserved									
0x28	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	
0x29 ... 0x2B	Reserved									
0x2C	DPLLRATIO	7:0	LDR[7:0]							
		15:8					LDR[11:8]			
		23:16					LDRFRAC[3:0]			
		31:24								
0x30	DPLLCTRLB	7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
		15:8				LBYPASS		LTIME[2:0]		
		23:16	DIV[7:0]							
		31:24						DIV[10:8]		
0x34	DPLLPRESC	7:0							PRESC[1:0]	
0x35 ... 0x37	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	DPLLSYNCBUSY	7:0					DPLLPRESC	DPLLRATIO	ENABLE	
0x39 ... 0x3B	Reserved									
0x3C	DPLLSTATUS	7:0							CLKRDY	LOCK

13.9.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to the 13.9.5.7 Register Access Protection section and the PAC - Peripheral Access Controller chapter for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" or "Write-Synchronized" property in each individual register description. Refer to the section on Synchronization for details.

13.9.8.1 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Loop Ratio Update Complete Interrupt Enable bit, which enables the DPLL Loop Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Timeout Interrupt Enable bit, which enables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Fall Interrupt Enable bit, which enables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Rise Interrupt Enable bit, which enables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Reference Clock Stopped Interrupt Enable bit, which enables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Lock Coarse Interrupt Enable bit, which enables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Lock Fine Interrupt Disable/Enable bit, disable the DFLL Lock Fine interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Out Of Bounds Interrupt Enable bit, which enables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Ready Interrupt Enable bit, which enables the DFLL Ready interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the OSC16M Ready Interrupt Enable bit, which enables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

13.9.8.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Timeout Interrupt Enable bit, which disables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Lock Coarse Interrupt Enable bit, which disables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Lock Fine Interrupt Enable bit, which disables the DFLL Lock Fine interrupt.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Out Of Bounds Interrupt Enable bit, which disables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the OSC16M Ready Interrupt Enable bit, which disables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.

Value	Description
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

13.9.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLLLDRTO) and will generate an interrupt request if INTENSET.DPLLLDRTO is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Loop Divider Ratio Update Complete interrupt flag.

Bit 18 – DPLLLTO DPLL Lock Timeout
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Timeout bit in the Status register (STATUS.DPLLLTO) and will generate an interrupt request if INTENSET.DPLLLTO is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Timeout interrupt flag.

Bit 17 – DPLLLCKF DPLL Lock Fall
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Fall bit in the Status register (STATUS.DPLLLCKF) and will generate an interrupt request if INTENSET.DPLLLCKF is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Fall interrupt flag.

Bit 16 – DPLLLCKR DPLL Lock Rise
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Rise bit in the Status register (STATUS.DPLLLCKR) and will generate an interrupt request if INTENSET.DPLLLCKR is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Rise interrupt flag.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 11 – DFLLLCKC DFLL Lock Coarse
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Lock Coarse bit in the Status register (STATUS.DFLLLCKC) and will generate an interrupt request if INTENSET.DFLLLCKC is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Lock Coarse interrupt flag.

Bit 10 – DFLLLCKF DFLL Lock Fine
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Lock Fine bit in the Status register (STATUS.DFLLLCKF) and will generate an interrupt request if INTENSET.DFLLLCKF is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Lock Fine interrupt flag.

Bit 9 – DFLLOOB DFLL Out Of Bounds
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Out Of Bounds bit in the Status register (STATUS.DFLLOOB) and will generate an interrupt request if INTENSET.DFLLOOB is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Out Of Bounds interrupt flag.

Bit 8 – DFLLRDY DFLL Ready
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Ready interrupt flag.

Bit 4 – OSC16MRDY OSC16M Ready
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the OSC16M Ready bit in the Status register (STATUS.OSC16MRDY) and will generate an interrupt request if INTENSET.OSC16MRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the OSC16M Ready interrupt flag.

Bit 0 – XOSCRDY XOSC Ready
This flag is cleared by writing '1' to it.
This flag is set on a 0-to-1 transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the XOSC Ready interrupt flag.

13.9.8.4 Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000100
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R	R	R	R	R
Reset				0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
Access				R		R	R	R
Reset				0		0	0	0

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete

Value	Description
0	DPLL Loop Divider Ratio Update Complete not detected.
1	DPLL Loop Divider Ratio Update Complete detected.

Bit 18 – DPLLLTO DPLL Lock Timeout

Value	Description
0	DPLL Lock time-out not detected.
1	DPLL Lock time-out detected.

Bit 17 – DPLLLCKF DPLL Lock Fall

Value	Description
0	DPLL Lock fall edge not detected.
1	DPLL Lock fall edge detected.

Bit 16 – DPLLLCKR DPLL Lock Rise

Value	Description
0	DPLL Lock rise edge not detected.
1	DPLL Lock fall edge detected.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 11 – DFLLLCKC DFLL Lock Coarse

Value	Description
0	No DFLL coarse lock detected.
1	DFLL coarse lock detected.

Bit 10 – DFLLLCKF DFLL Lock Fine

Value	Description
0	No DFLL fine lock detected.
1	DFLL fine lock detected.

Bit 9 – DFLLOOB DFLL Out Of Bounds

Value	Description
0	No DFLL Out Of Bounds detected.
1	DFLL Out Of Bounds detected.

Bit 8 – DFLLRDY DFLL Ready

Value	Description
0	DFLL registers update is ongoing. Registers update is requested through DFLLSYNC.READREQ, or after a write access in DFLLCTRL, DFLLVAL or DFLLMUL register.
1	DFLL registers are stable and ready for read/write access.

Bit 4 – OSC16MRDY OSC16M Ready

Value	Description
0	OSC16M is not ready.
1	OSC16M is stable and ready to be used as a clock source.

Bit 2 – CLKSW XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 1 – CLKFAIL XOSC Clock Failure

Value	Description
0	No XOSC failure detected.
1	A XOSC failure was detected.

Bit 0 – XOSCRDY XOSC Ready

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

13.9.8.5 16MHz Internal Oscillator (OSC16M) Control

Name: OSC16MCTRL
Offset: 0x14
Reset: 0x82
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	1	

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled depending on peripheral clock requests.
If the ONDEMAND bit has been previously written to '1', the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is disabled the oscillator will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the OSC16M behaves during standby sleep mode.

Value	Description
0	The OSC16M is disabled in standby sleep mode if no peripheral requests the clock.
1	The OSC16M is not stopped in standby sleep mode. If ONDEMAND=1, the OSC16M will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bits 3:2 – FSEL[1:0] Oscillator Frequency Selection
These bits control the oscillator frequency range.

Value	Description
0x00	4MHz
0x01	8MHz
0x10	12MHz
0x11	16MHz

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

13.9.8.6 External Multipurpose Crystal Oscillator (XOSC) Control

Name: XOSCCTRL
Offset: 0x10
Reset: 0x0080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	STARTUP[3:0]				AMPGC	GAIN[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				XTALEN	ENABLE	
Access	R/W	R/W				R/W	R/W	
Reset	1	0				0	0	

Bits 15:12 – STARTUP[3:0] Start-Up Time
These bits select start-up time for the oscillator.
The OSCULP32K oscillator is used to clock the start-up counter.

Table 13-31. Start-Up Time for External Multipurpose Crystal Oscillator

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0x0	1	3	31
0x1	2	3	61
0x2	4	3	122
0x3	8	3	244
0x4	16	3	488
0x5	32	3	977
0x6	64	3	1953
0x7	128	3	3906
0x8	256	3	7813
0x9	512	3	15625
0xA	1024	3	31250
0xB	2048	3	62500µs
0xC	4096	3	125000
0xD	8192	3	250000
0xE	16384	3	500000
0xF	32768	3	1000000

Notes:
1. Actual startup time is 1 OSCULP32K cycle + 3 XOSC cycles.
2. The given time neglects the three XOSC cycles before OSCULP32K cycle.

Bit 11 – AMPGC Automatic Amplitude Gain Control
Note: This bit must be set only after the XOSC has settled, indicated by the XOSC Ready flag in the Status register (STATUS.XOSCRDY).

Value	Description
0	The automatic amplitude gain control is disabled.
1	The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation.

Bits 10:8 – GAIN[2:0] Oscillator Gain
These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Those bits must be properly configured even when the Automatic Amplitude Gain Control is active.

Value	Recommended Max Frequency [MHz]
0x0	2
0x1	4
0x2	8
0x3	16
0x4	30
0x5-0x7	Reserved

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.
If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is disabled, the oscillator will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the XOSC behaves during standby sleep mode, together with the ONDEMAND bit:

Value	Description
0	The XOSC is not running in Standby sleep mode if no peripheral requests the clock.
1	The XOSC is running in Standby sleep mode. If ONDEMAND=1, the XOSC will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in Standby sleep mode.

Bit 2 – XTALEN Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

13.9.8.7 DFLL48M Control

Name: DFLLCTRL
Offset: 0x18
Reset: 0x0080
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	15	14	13	12	11	10	9	8
					WAITLOCK	BPLCKC	QLDIS	CCDIS
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	0	0	0	0	0	0	

Bit 11 – WAITLOCK Wait Lock
This bit controls the DFLL output clock, depending on lock status.

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

Bit 10 – BPLCKC Bypass Coarse Lock
This bit controls the coarse lock procedure.

Value	Description
0	Bypass coarse lock is disabled.
1	Bypass coarse lock is enabled.

Bit 9 – QLDIS Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 8 – CCDIS Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests. If the ONDEMAND bit has been previously written to '1', the DFLL will only be running when requested by a peripheral. If there is no peripheral requesting the DFLL clock source, the DFLL will be in a disabled state.
If On Demand is disabled, the DFLL will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DFLL is always on, if enabled.
1	The DFLL is enabled when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the DFLL behaves during standby sleep mode:

Value	Description
0	The DFLL is disabled in standby sleep mode if no peripheral requests the clock.

Value	Description
1	The DFLL is not stopped in standby sleep mode. If ONDEMAND is one, the DFLL will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

Bit 5 – USBCRM USB Clock Recovery Mode

Value	Description
0	USB Clock Recovery Mode is disabled.
1	USB Clock Recovery Mode is enabled.

Bit 4 – LLAW Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

Bit 3 – STABLE Stable DFLL Frequency

Value	Description
0	FINE calibration tracks changes in output frequency.
1	FINE calibration register value will be fixed after a fine lock.

Bit 2 – MODE Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

Bit 1 – ENABLE DFLL Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRL.ENABLE will read back immediately after written.

Value	Description
0	The DFLL oscillator is disabled.
1	The DFLL oscillator is enabled.

13.9.8.8 DFLL48M Value

Name: DFLLVAL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Read-Synchronized using DFLLSYNC.READREQ, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	DIFF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIFF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COARSE[5:0]						FINE[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FINE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DIFF[15:0] Multiplication Ratio Difference
In closed-loop mode (DFLLCTRL.MODE=1), this bit group indicates the difference between the ideal number of DFLL cycles and the counted number of cycles. In open-loop mode, this value is not updated and hence, invalid.

Bits 15:10 – COARSE[5:0] Coarse Value
Set the value of the Coarse Calibration register. In closed-loop mode, this field is read-only.

Bits 9:0 – FINE[9:0] Fine Value
Set the value of the Fine Calibration register. In closed-loop mode, this field is read-only.

13.9.8.9 DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	CSTEP[5:0]						FSTEP[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MUL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – CSTEP[5:0] Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 25:16 – FSTEP[9:0] Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 15:0 – MUL[15:0] DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.

13.9.8.10 DFLL48M Synchronization

Name: DFLLSYNC
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	READREQ							
Access	W							
Reset	0							

Bit 7 – READREQ Read Request
To be able to read the current value of the DFLLVAL register in closed-loop mode, this bit must be written to '1'.

13.9.8.11 DPLL Control A

Name: DPLLCTRLA
Offset: 0x28
Reset: 0x80
Property: PAC Write-Protection, Write-Synchronized (ENABLE)

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	
Access	R/W	R/W					R/W	
Reset	1	0					0	

Bit 7 – ONDEMAND On Demand Clock Activation
The On Demand operation mode allows the DPLL to be enabled or disabled depending on peripheral clock requests. If the ONDEMAND bit has been previously written to '1', the DPLL will only be running when requested by a peripheral. If there is no peripheral requesting the DPLL's clock source, the DPLL will be in a disabled state.
If On Demand is disabled the DPLL will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DPLL is always on, if enabled.
1	The DPLL is enabled when a peripheral is requesting the DPLL to be used as a clock source. The DPLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the DPLL behaves during standby sleep mode:

Value	Description
0	The DPLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DPLL is not stopped in standby sleep mode. If ONDEMAND=1, the DPLL will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bit 1 – ENABLE DPLL Enable
The software operation of enabling or disabling the DPLL takes a few clock cycles, so the DPLLSYNCBUSY.ENABLE status bit indicates when the DPLL is successfully enabled or disabled.

Value	Description
0	The DPLL is disabled.
1	The DPLL is enabled.

13.9.8.12 DPLL Ratio Control

Name: DPLLRATIO
Offset: 0x2C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LDRFRAC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LDR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – LDRFRAC[3:0] Loop Divider Ratio Fractional Part
Writing these bits selects the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing these bits and the effect on the DPLL output clock. The value written will read back immediately and the DPLLRATIO bit in the DPLL Synchronization Busy register (DPLLSYNCBUSY.DPLLRATIO) will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Bits 11:0 – LDR[11:0] Loop Divider Ratio
Writing these bits selects the integer part of the frequency multiplier. The value written to these bits will read back immediately, and the DPLLRATIO bit in the DPLL Synchronization busy register (DPLLSYNCBUSY.DPLLRATIO), will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

13.9.8.13 DPLL Control B

Name: DPLLCTRLB
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						DIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				LBYPASS		LTIME[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 26:16 – DIV[10:0] Clock Divider
These bits set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f_{XOSC}}{2x(DIV + 1)}$$

Bit 12 – LBYPASS Lock Bypass

Value	Description
0	DPLL Lock signal drives the DPLL controller internal logic.
1	DPLL Lock signal is always asserted.

Bits 10:8 – LTIME[2:0] Lock Time
These bits select the lock time-out value:

Value	Name	Description
0x0	Default	No time-out. Automatic lock.
0x1	Reserved	
0x2	Reserved	
0x3	Reserved	
0x4	8MS	Time-out if no lock within 8ms
0x5	9MS	Time-out if no lock within 9ms
0x6	10MS	Time-out if no lock within 10ms
0x7	11MS	Time-out if no lock within 11ms

Bits 5:4 – REFCLK[1:0] Reference Clock Selection
Write these bits to select the DPLL clock reference:

Value	Name	Description
0x0	XOSC32K	XOSC32K clock reference
0x1	XOSC	XOSC clock reference
0x2	GCLK	GCLK clock reference
0x3	Reserved	

Bit 3 – WUF Wake Up Fast

Value	Description
0	DPLL clock is output after startup and lock time.
1	DPLL clock is output after startup time.

Bit 2 – LPEN Low-Power Enable

Value	Description
0	The low-power mode is disabled. Time to Digital Converter is enabled.
1	The low-power mode is enabled. Time to Digital Converter is disabled. This will improve power consumption but increase the output jitter.

Bits 1:0 – FILTER[1:0] Proportional Integral Filter Selection
These bits select the DPLL filter type:

Value	Name	Description
0x0	DEFAULT	Default filter mode
0x1	LBFILT	Low bandwidth filter
0x2	HBFILT	High bandwidth filter
0x3	HDFILT	High damping filter

13.9.8.14 DPLL Prescaler

Name: DPLLPRESC
Offset: 0x34
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							PRESC[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – PRESC[1:0] Output Clock Prescaler
These bits define the output clock prescaler setting.

Value	Name	Description
0x0	DIV1	DPLL output is divided by 1
0x1	DIV2	DPLL output is divided by 2
0x2	DIV4	DPLL output is divided by 4
0x3	Reserved	

13.9.8.15 DPLL Synchronization Busy

Name: DPLLSYNCBUSY
Offset: 0x38
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					DPLLPRESC	DPLLRATIO	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DPLLPRESC DPLL Prescaler Synchronization Status

Value	Description
0	The DPLLRESC register has been synchronized.
1	The DPLLRESC register value has changed and its synchronization is in progress.

Bit 2 – DPLLRATIO DPLL Loop Divider Ratio Synchronization Status

Value	Description
0	The DPLLRATIO register has been synchronized.
1	The DPLLRATIO register value has changed and its synchronization is in progress.

Bit 1 – ENABLE DPLL Enable Synchronization Status

Value	Description
0	The DPLLCTRLA.ENABLE bit has been synchronized.
1	The DPLLCTRLA.ENABLE bit value has changed and its synchronization is in progress.

13.9.8.16 DPLL Status

Name: DPLLSTATUS
Offset: 0x3C
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							CLKRDY	LOCK
Access							R	R
Reset							0	0

Bit 1 – CLKRDY Output Clock Ready

Value	Description
0	The DPLL output clock is off.
1	The DPLL output clock in on.

Bit 0 – LOCK DPLL Lock status bit

Value	Description
0	The DPLL Lock signal is cleared, when the DPLL is disabled or when the DPLL is trying to reach the target frequency.
1	The DPLL Lock signal is asserted when the desired frequency is reached.

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.10.1 Overview

The 32KHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768kHz oscillators: XOSC32K, OSC32K, and OSCULP32K.

The OSC32KCTRL sub-peripherals can be enabled, disabled, calibrated, and monitored through interface registers.

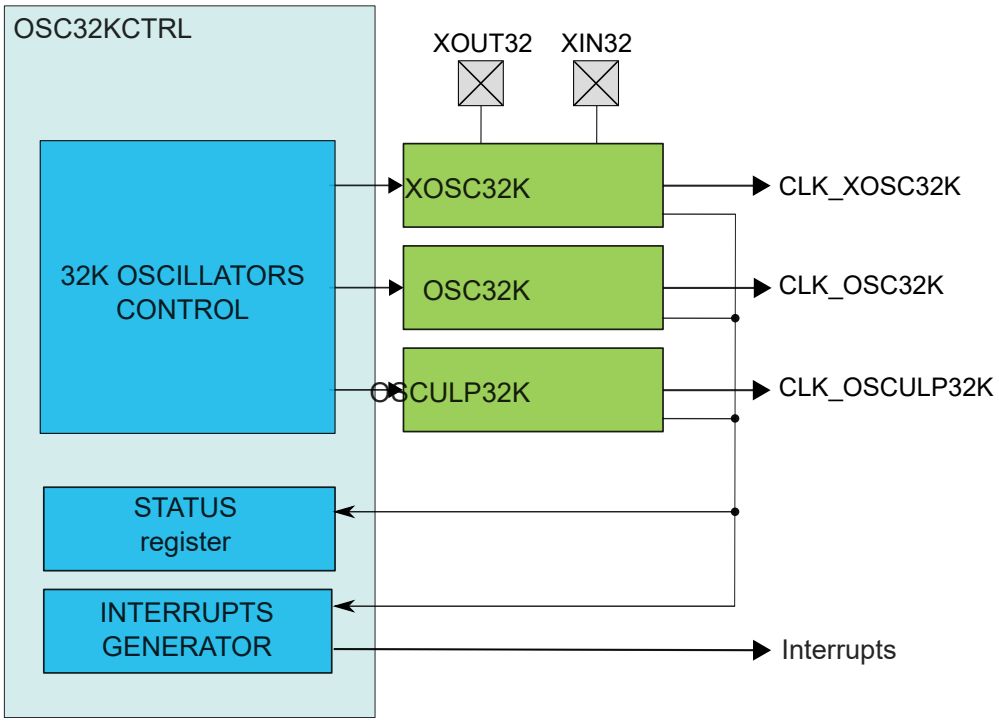
All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

13.10.2 Features

- 32.768kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
- 32.768kHz High Accuracy Internal Oscillator (OSC32K)
  - Frequency fine tuning
  - Programmable start-up time
- 32.768kHz Ultra Low Power Internal Oscillator (OSCULP32K)
  - Ultra low power, always-on oscillator
  - Frequency fine tuning
- Calibration value loaded from Flash factory calibration at reset
- 1.024kHz clock outputs available

13.10.3 Block Diagram

Figure 13-37. OSC32KCTRL Block Diagram



13.10.4 Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768kHz Crystal Oscillator or external clock generator input
XOUT32	Analog Output	32.768kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled.

Note: The signal of the external crystal oscillator may affect the jitter of neighboring pads.

13.10.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.10.5.1 I/O Lines

I/O lines are configured by OSC32KCTRL when XOSC32K is enabled, and need no user configuration.

13.10.5.2 Power Management

The OSC32KCTRL will continue to operate in any sleep mode where a 32KHz oscillator is running as source clock. The OSC32KCTRL interrupts can be used to wake up the device from sleep modes.

Related Links

13.8 PM – Power Manager

13.10.5.3 Clocks

The OSC32KCTRL gathers controls for all 32KHz oscillators and provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), and Watchdog Timer (WDT).

The available clock sources are: XOSC32K, OSC32K, and OSCULP32K.

The OSC32KCTRL bus clock (CLK_OSC32KCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.10.5.4 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the OSC32KCTRL interrupts requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.10.5.5 Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.10.5.6 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.10.5.7 Analog Connections

The external 32.768kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the related links.

Related Links

15. Electrical Characteristics

13.10.5.8 Calibration

The OSC32K calibration value from the production test must be loaded from the NVM Software Calibration Area into the OSC32K register (OSC32K.CALIB) by software to achieve specified accuracy.

13.10.6 Functional Description

13.10.6.1 Principle of Operation

XOSC32K, OSC32K, and OSCULP32K are configured via OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

13.10.6.2 32 KHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768 kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and the GPIO functions are overridden on both pins. When in external clock mode, the only XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The XOSC32K is enabled by writing a '1' to the Enable bit in the 32 KHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 1). The XOSC32K is disabled by writing a '0' to the Enable bit in the 32 KHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 0).

To enable the XOSC32K as a crystal oscillator, the XTALEN bit in the 32 KHz External Crystal Oscillator Control register must be set (XOSC32K.XTALEN = 1). If XOSC32K.XTALEN is '0', the external clock input will be enabled.

The XOSC32K 32.768 kHz output is enabled by setting the 32 KHz Output Enable bit in the 32 KHz External Crystal Oscillator Control register (XOSC32K.EN32K = 1). The XOSC32K also has a 1.024 kHz clock output, which can only be used by the RTC. This clock output is enabled by setting the 1 KHz Output Enable bit in the 32 KHz External Crystal Oscillator Control register (XOSC32K.EN1K = 1).

It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32 KHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK = 1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND and XOSC32K.ENABLE. If XOSC32K.ENABLE = 0, the XOSC32K will always be stopped. For XOS32K.ENABLE = 1, this table is valid:

Table 13-32. XOSC32K Sleep Behavior

CPU Mode	XOSC32K. RUNSTDBY	XOSC32K. ONDEMAND	Sleep Behavior
Active or Idle	—	0	Always run
Active or Idle	—	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	—	Run if requested by peripheral

As a crystal oscillator usually requires a very long start-up time, the 32 KHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND = 0, except for power-on reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32 KHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY = 1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY = 1).

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed. For details on the RTC clock configuration, refer to 13.10.6.6 Real-Time Counter Clock Selection.

13.10.6.3 32 KHz Internal Oscillator (OSC32K) Operation

The OSC32K provides a tunable, low-speed and low-power clock source.

At reset, the OSC32K is disabled. It can be enabled by setting the Enable bit in the 32 KHz Internal Oscillator Control register (OSC32K.ENABLE = 1). The OSC32K is disabled by clearing the Enable bit in the 32 KHz Internal Oscillator Control register (OSC32K.ENABLE = 0).

The frequency of the OSC32K oscillator is controlled by OSC32K.CALIB, which is a calibration value in the 32 KHz Internal Oscillator Calibration bits in the 32 KHz Internal Oscillator Control register. The CALIB value must be loaded with production calibration values from the NVM Software Calibration Area. When writing the Calibration bits, the user must wait for the STATUS.OSC32KRDY bit to go high before the new value is committed to the oscillator.

The OSC32K has a 32.768 kHz output which is enabled by setting the 32 KHz Output Enable bit in the 32 KHz Internal Oscillator Control register (OSC32K.EN32K = 1). The OSC32K also has a 1.024 kHz clock output. This is enabled by setting the 1 KHz Output Enable bit in the 32 KHz Internal Oscillator Control register (OSC32K.EN1K).

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area

The OSC32K will behave differently in different sleep modes based on the settings of OSC32K.RUNSTDBY, OSC32K.ONDEMAND and OSC32K.ENABLE. If OSC32KCTRL.ENABLE = 0, the OSC32K will always be stopped. For OS32K.ENABLE = 1, this table is valid:

Table 13-33. OSC32K Sleep Behavior

CPU Mode	OSC32K.RUNSTDBY	OSC32K.ONDEMAND	Sleep Behavior
Active or Idle	—	0	Always run
Active or Idle	—	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	—	Run if requested by peripheral

The OSC32K requires a start-up time. For this reason, OSC32K will keep running across resets when OSC32K.ONDEMAND = 0, except for power-on reset (POR).

After such a reset, or when waking up from a sleep mode where the OSC32K was disabled, the OSC32K will need a certain amount of time to stabilize on the correct frequency.

This start-up time can be configured by changing the Oscillator Start-Up Time bit group (OSC32K.STARTUP) in the OSC32K Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the OSC32K Ready bit in the Status register is set (STATUS.OSC32KRDY = 1). The transition of STATUS.OSC32KRDY from '0' to '1' generates an interrupt if the OSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.OSC32KRDY = 1).

The OSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (OSC32K.EN32K or OSC32K.EN1K) to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed.

13.10.6.4 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra-low-power clock source. The OSCULP32K is factory-calibrated under typical voltage and temperature conditions. The OSCULP32K should be preferred to the OSC32K whenever the power requirements are prevalent over frequency stability and accuracy.

The OSCULP32K is enabled by default after a power-on reset (POR) and will always run except during POR. The frequency of the OSCULP32K oscillator is controlled by the value in the 32KHz Ultra Low Power Internal Oscillator Calibration bits in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.CALIB). This data is used to compensate for process variations.

OSCULP32K.CALIB is automatically loaded from Flash Factory Calibration during start-up. The calibration value can be overridden by the user by writing to OSCULP32K.CALIB.

It is also possible to lock the OSCULP32K configuration by setting the Write Lock bit in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.WRTLOCK=1). If set, the OSCULP32K configuration is locked until a power-on reset (POR) is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

Related Links

13.13 RTC – Real-Time Counter

13.10.6.6 Real-Time Counter Clock Selection

13.5 GCLK - Generic Clock Controller

13.10.6.5 Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

Related Links

13.12 WDT – Watchdog Timer

13.10.6.6 Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

Related Links

13.13 RTC – Real-Time Counter

13.10.6.7 Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32KHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- OSC32KRDY - 32KHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC32KRDY bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

13.8 PM – Power Manager

11.2 Nested Vector Interrupt Controller

13.10.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	7:0							OSC32KRDY	XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x04	INTENSET	7:0							OSC32KRDY	XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x08	INTFLAG	7:0							OSC32KRDY	XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x0C	STATUS	7:0							OSC32KRDY	XOSC32KRDY
		15:8								
		23:16								
		31:24								
0x10	RTCCTRL	7:0						RTCSEL[2:0]		
		15:8								
		23:16								
		31:24								
0x14	XOSC32K	7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
		15:8				WRTLOCK		STARTUP[2:0]		
		23:16								
		31:24								
0x18	OSC32K	7:0	ONDEMAND	RUNSTDBY			EN1K	EN32K	ENABLE	
		15:8				WRTLOCK		STARTUP[2:0]		
		23:16		CALIB[6:0]						
		31:24								
0x1C	OSCULP32K	7:0								
		15:8	WRTLOCK			CALIB[4:0]				
		23:16								
		31:24								

13.10.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in the register description. Write-protection does not apply to accesses through an external debugger.

13.10.8.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY OSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the OSC32K Ready Interrupt Enable bit, which disables the OSC32K Ready interrupt.

Value	Description
0	The OSC32K Ready interrupt is disabled.
1	The OSC32K Ready interrupt is enabled.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

13.10.8.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY OSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the OSC32K Ready Interrupt Enable bit, which enables the OSC32K Ready interrupt.

Value	Description
0	The OSC32K Ready interrupt is disabled.
1	The OSC32K Ready interrupt is enabled.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

13.10.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY OSC32K Ready
This flag is cleared by writing a '1' to it.
This flag is set by a zero-to-one transition of the OSC32K Ready bit in the Status register (STATUS.OSC32KRDY), and will generate an interrupt request if INTENSET.OSC32KRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the OSC32K Ready interrupt flag.

Bit 0 – XOSC32KRDY XOSC32K Ready
This flag is cleared by writing a '1' to it.
This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.

13.10.8.4 Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R	R
Reset							0	0

Bit 1 – OSC32KRDY OSC32K Ready

Value	Description
0	OSC32K is not ready.
1	OSC32K is stable and ready to be used as a clock source.

Bit 0 – XOSC32KRDY XOSC32K Ready

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

13.10.8.5 RTC Clock Selection Control

Name: RTCCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RTCSEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – RTCSEL[2:0] RTC Clock Source Selection
These bits select the source for the RTC.

Value	Name	Description
0x0	ULP1K	1.024kHz from 32KHz internal ULP oscillator
0x1	ULP32K	32.768kHz from 32KHz internal ULP oscillator
0x2	OSC1K	1.024kHz from 32KHz internal oscillator
0x3	OSC32K	32.768kHz from 32KHz internal oscillator
0x4	XOSC1K	1.024kHz from 32KHz external oscillator
0x5	XOSC32K	32.768kHz from 32KHz external crystal oscillator
0x6	Reserved	
0x7	Reserved	

13.10.8.6 32 KHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x14
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bit 12 – WRTLOCK Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

Value	Description
0	The XOSC32K configuration is not locked.
1	The XOSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0] Oscillator Start-Up Time
These bits select the start-up time for the oscillator.
The OSCULP32K oscillator is used to clock the start-up counter.

Table 13-34. Start-Up Time for 32 KHz External Crystal Oscillator

STARTUP[2:0]	Number of OSCULP32K Clock Cycles	Number of XOSC32K Clock Cycles	Approximate Equivalent Time [s]
0x0	2048	3	0.06
0x1	4096	3	0.13
0x2	16384	3	0.5
0x3	32768	3	1
0x4	65536	3	2
0x5	131072	3	4
0x6	262144	3	8
0x7	—	—	Reserved

Notes:

1. Actual Start-Up time is 1 OSCULP32K cycle + 3 XOSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768 kHz.

Bit 7 – ONDEMAND On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to Table 13-32.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to Table 13-32.

Bit 4 – EN1K 1 KHz Output Enable

Value	Description
0	The 1 KHz output is disabled.
1	The 1 KHz output is enabled.

Bit 3 – EN32K 32 KHz Output Enable

Value	Description
0	The 32 KHz output is disabled.
1	The 32 KHz output is enabled.

Bit 2 – XTALEN Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

13.10.8.7 32 KHz Internal Oscillator (OSC32K) Control

Name: OSC32K
Offset: 0x18
Reset: 0x0000 0080 (Writing action by User required)
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CALIB[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			EN1K	EN32K	ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	0	

Bits 22:16 – CALIB[6:0] Oscillator Calibration
These bits control the oscillator calibration. The calibration values must be loaded by the user from the NVM Software Calibration Area.

Bit 12 – WRTLOCK Write Lock
This bit locks the OSC32K register for future writes, effectively freezing the OSC32K configuration.

Value	Description
0	The OSC32K configuration is not locked.
1	The OSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0] Oscillator Start-Up Time
These bits select start-up time for the oscillator.
The OSCULP32K oscillator is used as input clock to the start-up counter.

Table 13-35. Start-Up Time for 32KHz Internal Oscillator

STARTUP[2:0]	Number of OSC32K clock cycles	Approximate Equivalent Time [ms]
0x0	3	0.092
0x1	4	0.122
0x2	6	0.183
0x3	10	0.305
0x4	18	0.549
0x5	34	1.038
0x6	66	2.014
0x7	130	3.967

Notes:
1. Start-up time is given by STARTUP + three OSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768 kHz.

Bit 7 – ONDEMAND On Demand Control
This bit controls how the OSC32K behaves when a peripheral clock request is detected. For details, refer to Table 13-32.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the OSC32K behaves during standby sleep mode. For details, refer to Table 13-32.

Bit 3 – EN1K 1 KHz Output Enable

Value	Description
0	The 1 KHz output is disabled.
1	The 1 KHz output is enabled.

Bit 2 – EN32K 32 KHz Output Enable

Value	Description
0	The 32 KHz output is disabled.
1	The 32 KHz output is enabled.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

13.10.8.8 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Control

Name: OSCULP32K
Offset: 0x1C
Reset: 0x0000XX06
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WRTLOCK			CALIB[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – WRTLOCK Write Lock
This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

Bits 12:8 – CALIB[4:0] Oscillator Calibration
These bits control the oscillator calibration.
These bits are loaded from Flash Calibration at startup.

13.11 SUPC – Supply Controller

13.11.1 Overview

The Supply Controller (SUPC) manages the voltage reference, power supply, and supply monitoring of the device. It is also able to control two output pins.

The SUPC controls the voltage regulators for the core (VDDCORE) and backup (VDDBU) domains. It sets the voltage regulators according to the sleep modes, or the user configuration.

The SUPC supports connection of a battery backup to the VBAT power pin. It includes functionality that enables automatic power switching between main power and battery backup power. This ensures power to the backup domain when the main battery or power source is unavailable.

The SUPC embeds two Brown-Out Detectors. BOD33 monitors the voltage applied to the device (VDD or VBAT) and BOD12 monitors the internal voltage to the core (VDDCORE). The BOD can monitor the supply voltage continuously (continuous mode) or periodically (sampling mode).

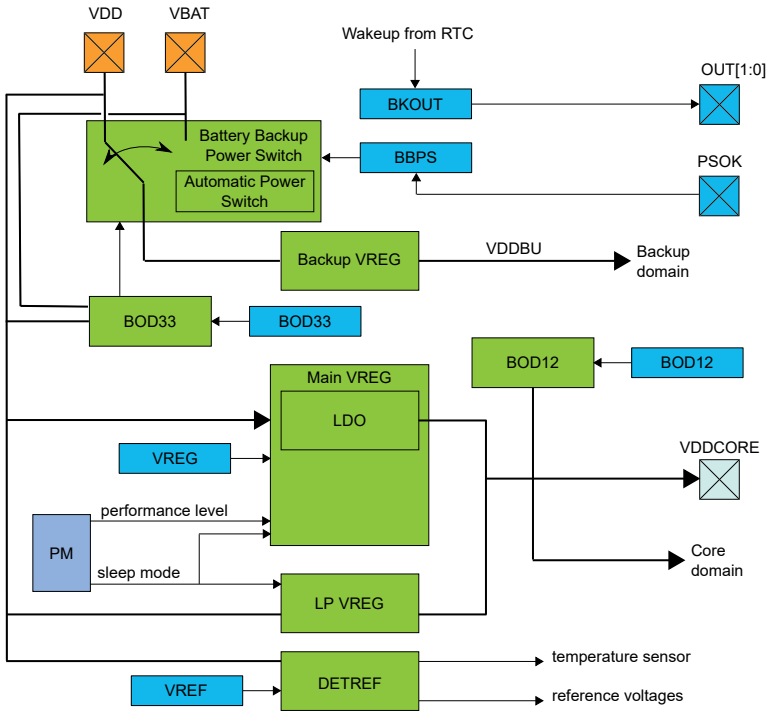
The SUPC generates also a selectable reference voltage and a voltage dependent on the temperature which can be used by analog modules like the ADC.

13.11.2 Features

- Voltage Regulator System
  - Main voltage regulator: LDO in active mode (MAINVREG)
  - Low Power voltage regulator in standby mode (LPVREG)
  - Backup voltage regulator for backup domains
  - Controlled VDDCORE voltage slope when changing VDDCORE
- Battery Backup Power Switch
  - Automatic switching from main power to battery backup power
    - Automatic entry to backup mode when switched to battery backup power
  - Automatic switching from battery backup power to main power
    - Automatic exit from backup mode when switched back to main power
    - Stay in backup mode when switched back to main power
  - Main power request upon wake-up sources from backup mode
- Voltage Reference System
  - Reference voltage for ADC
  - Temperature sensor
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at startup
  - Triggers resets, interrupts, or Battery Backup Power Switch. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode
    - Sampled mode for low power applications with programmable sample frequency
  - Hysteresis value from Flash User Calibration
  - Monitor VDD or VBAT
- 1.2V Brown-Out Detector (BOD12)
  - Internal non-configurable Brown-Out Detector
- Output pins
  - Pin toggling on RTC event

13.11.3 Block Diagram

Figure 13-38. SUPC Block Diagram



13.11.4 Signal Description

Signal Name	Type	Description
OUT[1:0]	Digital Output	SUPC Outputs
PSOK	Digital Input	Main Power Supply OK

One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.11.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.11.5.1 I/O Lines

I/O lines are configured by SUPC either when the SUPC output (signal OUT) is enabled or when the PSOK input is enabled. The I/O lines need no user configuration.

13.11.5.2 Power Management

The SUPC can operate in all sleep modes except backup sleep mode. BOD33 and Battery backup Power Switch can operate in backup mode.

Related Links

13.8 PM – Power Manager

13.11.5.3 Clocks

The SUPC bus clock (CLK_SUPC_APB) can be enabled and disabled in the Main Clock module.

A 32KHz clock, asynchronous to the user interface clock (CLK_SUPC_APB), is required to run BOD33 and BOD12 in sampled mode. Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 13.11.6.7 Synchronization for further details.

Related Links

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.6.6.2.6 Peripheral Clock Masking

13.11.5.4 DMA

Not applicable.

13.11.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the SUPC interrupts requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.11.5.6 Events

Not applicable.

13.11.5.7 Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

If debugger cold-plugging is detected by the system, BOD33 and BOD12 resets will be masked. The BOD resets keep running under hot-plugging. This allows to correct a BOD33 user level too high for the available supply.

13.11.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

Note: Not all registers with write-access can be write-protected.

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

13.11.5.9 Analog Connections

Not applicable.

13.11.6 Functional Description

13.11.6.1 Voltage Regulator System Operation

13.11.6.1.1 Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after any Reset. The main voltage regulator (MAINVREG) can be disabled by writing the Enable bit in the VREG register (VREG.ENABLE) to zero. The main voltage regulator output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

Related Links

13.8 PM – Power Manager

13.11.6.1.2 Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled.

13.11.6.1.3 Voltage Scaling Control

The VDDCORE supply will change under certain circumstances:

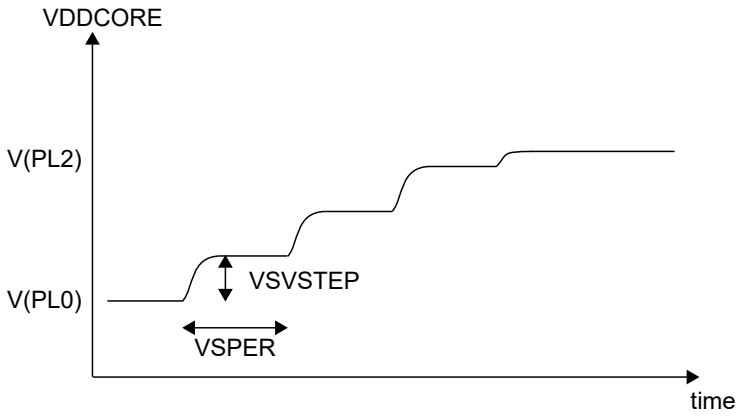
- When a new performance level (PL) is set

- When the standby sleep mode is entered or left
- When a sleepwalking task is requested in standby sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also 13.11.6.6 Interrupts.

When entering the standby sleep mode and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used.

13.11.6.1.4 Sleep Mode Operation

In standby mode, the low power voltage regulator (LPVREG) is used to supply VDDCORE.

When the Run in Standby bit in the VREG register (VREG.RUNSTDBY) is written to '1', VDDCORE is supplied by the main voltage regulator. Depending on the Standby in PL0 bit in the Voltage Regulator register (VREG.STDBYPL0), the VDDCORE level is either set to the PL0 voltage level, or remains in the current performance level.

Table 13-36. VDDCORE Level in Standby Mode

VREG.RUNSTDBY	VREG.STDBYPL0	VDDCORE Supply in Standby Mode
0	-	LPVREG
1	0	MAINVREG in current performance level(1)
1	1	MAINVREG in PL0

Note:

1. When the device is in PL0 but VREG.STDBYPL0=0, the MAINVREG is operating in normal power mode. To minimize power consumption, operate MAINVREG in PL0 mode by selecting VREG.STDBYPL0=1.

13.11.6.2 Voltage Reference System Operation

The reference voltages are generated by a functional block DETREF inside of the SUPC. DETREF is providing a fixed-voltage source, BANDGAP=1V, and a variable voltage, INTREF.

13.11.6.2.1 Initialization

The voltage reference output and the temperature sensor are disabled after any Reset.

13.11.6.2.2 Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREF.TSEN).

Note: When VREF.ONDEMAND=0, it is not recommended to enable both voltage reference output and temperature sensor at the same time - only the voltage reference output will be present at both ADC inputs.

13.11.6.2.3 Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of INTREF to be applied to analog modules, e.g. the ADC.

13.11.6.2.4 Sleep Mode Operation

The Voltage Reference output and the Temperature Sensor output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 13-37. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes *except* standby sleep mode
0	1	Always run in all sleep modes *including* standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes *except* standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes *including* standby sleep mode

13.11.6.3 Battery Backup Power Switch

13.11.6.3.1 Initialization

The Battery Backup Power Switch (BBPS) is disabled at power-up, and the backup domain is supplied by main power.

13.11.6.3.2 Forced Battery Backup Power Switch

The Backup domain is always supplied by the VBAT supply pin when the Configuration bit field in the Battery Backup Power Switch Control register (BBPS.CONF) is written to 0x2 (FORCED).

13.11.6.3.3 Automatic Battery Backup Power Switch

The supply of the backup domain can be switched automatically to VBAT supply pin by the Automatic Power Switch or by using the BOD33.

The supply of the backup domain can be switched automatically to VDD supply pin either by the Automatic Power Switch or the Main Power Pin when VDD and VDDCORE are restored.

Automatic Power Switch (APWS)

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) is selecting the APWS, the Automatic Power Switch will function as Battery Backup Power Switch.

The Automatic Power switch allows to switch the supply of the backup domain from VDD to VBAT power and vice-versa.

When the Automatic Power Switch configuration is selected, the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) is set when the Automatic Power Switch is ready to operate. The Automatic Power Switch Ready bit in the Interrupt Flag Status and Clear (INTFLAG.APSWRDY) will be set at the same time.

Related Links

15. Electrical Characteristics

BOD33 Power Switch
When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) are selecting the BOD33, BOD33 will function as Battery Backup Power Switch. In this case, when the VDD voltage is below the BOD33 threshold, the backup domain supply is switched to VBAT.

Main Power Supply OK (PSOK) Pin Enable
The state of the Main Power VDD can be used to switch between supply sources as long as the Battery Backup Power Switch is not configured as Automatic Power Switch (i.e., BBPS.CONF not set to APWS): when the Main Power Supply OK Pin Enable bit in the BBPS register is written to '1' (BBPS.PSOKEN), restoring VDD will form a low-to-high transition on the PSOK pin. This low-to-high transition will switch the Backup Power Supply back to VDD.

Note: With BBPS.PSOKEN=0 and BBPS.CONF not configured to APWS, the device can not be restarted.

Backup Battery Power Switch Status
The Battery Backup Power Switch bit in the Status register (STATUS.BBPS) indicates whether the backup domain is currently powered by VDD or VBAT.

13.11.6.3.4 Sleep Mode Operation
The Battery Backup Power Switch is not stopped in any sleep mode.

Entering Battery Backup Mode
Entering backup mode can be triggered by either:

- Wait-for-interrupt (WFI) instruction.
- Automatic Power Switch (BBPS.CONF=APWS). When the Automatic Power Switch detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode.
- BOD33 detection: When the BOD33 detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode. For this trigger, the following register configuration is required: BOD33.ACTION=BKUP, BOD33.VMON=VDD, and BBPS.CONF=BOD33.

Related Links
13.8 PM – Power Manager

Leaving Battery Backup Mode
Leaving backup mode can be triggered by either:

- RTC requests and externally triggered RSTC requests, under one of these conditions:
  - The Backup Domain is supplied by Main Power, and the Battery Backup Power Switch is *not* forced (BBPS.CONF not set to FORCED)
  - The Battery Backup Power Switch *is* forced (BBPS.CONF is FORCED)

  The device is kept in battery-powered backup mode until Main Power is restored to supply the device. Then, the backup domain will be powered by Main Power.
- Automatic Power Switch. Leaving backup mode will happen when Main Power is restored and the Battery Backup Power Switch configuration (BBPS.CONF) is set to APWS: When BBPS.WAKEEN=1, the device will leave backup mode and wake up.
  When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.
- PSOK pin. A low-to-high transition on PSOK will wake up the device if BBPS.PSOKEN=1, BBPS.WAKEEN=1, and the Battery Backup Power Switch is different from APWS (BBPS.CONF is not APWS).
  When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.

13.11.6.4 Output Pins
The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the OUTx pin is driven by the SUPC.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx).

The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If a RTC Toggle Enable bit is written to '1' (BKOUT.RTCTGLx), the corresponding OUTx pin will toggle when an RTC event occurs.

13.11.6.5 Brown-Out Detectors

13.11.6.5.1 Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in Active, Standby, Backup modes (BOD33.ACTCDG, BOD33.STDBYCFG, BODVDD.BKUP)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected, meaning that it can only be written when the BOD is disabled (BOD33.ENABLE = 0 and SYNCBUSY.BOD33EN = 0). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

13.11.6.5.2 Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Row.

The BODVDD is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD is disabled by writing a '0' to the BODVDD.ENABLE.

Related Links

10.3 NVM User Row Mapping

13.11.6.5.3 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor either the VDD or the VBAT supply .

The Voltage Monitored bit in the BOD33 Control register (BOD33.VMON) selects which supply is monitored in active and standby mode. In backup mode, BOD33 will always monitor the supply of the backup domain, i.e. either VDD or VBAT.

If VDD is monitored, the BOD33 compares the voltage with the brown-out threshold level. This level is set in the BOD33 Level field in the BOD33 register (BOD33.LEVEL). This level is used in all modes except the backup sleep modes. In backup sleep modes, a different voltage reference is used, which is configured by the BOD33.BKUPLEVEL bits.

When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt, a Reset, or an Automatic Battery Backup Power Switch, depending on the BOD33 Action bit field (BOD33.ACTION).

If VBAT is monitored, the BOD33 compares the voltage with the brown-out threshold level set in the BOD33 Backup Level field in the BOD33 register (BOD33.BKUPLEVEL).

When VBAT crosses below the backup brown-out threshold level, the BOD33 can generate either an interrupt or a Reset.

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

Related Links

10.3 NVM User Row Mapping

13.11.6.5.4 1.2V Brown-Out Detector (BOD12)

The BOD12 is calibrated in production and its calibration configuration is stored in the NVM User Row. This configuration must not be changed to assure the correct behavior of the BOD12. The BOD12 generates a reset when 1.2V crosses below the preset brown-out level. The BODCORE is always disabled in standby sleep mode.

Related Links

10.3 NVM User Row Mapping

13.11.6.5.5 Continuous Mode

Continuous mode is the default mode for BOD33.

The BOD33 is continuously monitoring the supply voltage (VDD or VBAT, depending on BOD33.VMON) if it is enabled (BOD33.ENABLE=1) and if the BOD33 Configuration bit in the BOD33 register is cleared (BOD33.ACTCFG=0 for active mode, BOD33.STDBYCFG=0 for standby mode).

13.11.6.5.6 Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 is being repeatedly enabled on a sampling clock's ticks. The BOD33 will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Active mode for BOD33 by writing the ACTCFG bit (BOD33.ACTCFG=1). Sampling mode is enabled in Standby mode by writing to the STDBYCFG bit (BOD33.STBYCFG=1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the BOD33 register (BOD33.PSEL).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 1KHz clock, output by the 32KHz Ultra Low Power Oscillator OSCULP32K.

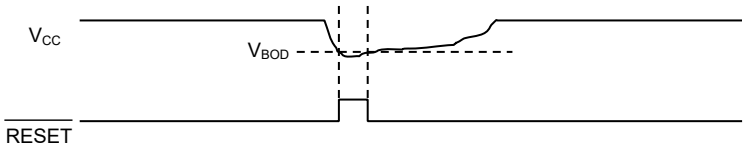
As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See also 13.11.6.7 Synchronization.

13.11.6.5.7 Hysteresis

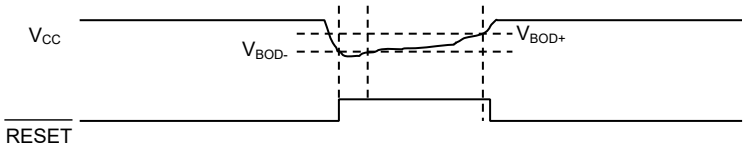
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of V_{BOD}, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated (V_{BOD-} and V_{BOD+}, respectively).

Figure 13-39. BOD Hysteresis Principle

Hysteresis OFF:



Hysteresis ON:



Enabling the BOD33 hysteresis by writing the Hysteresis bit in the BOD33 register (BOD33.HYST) to '1' will add hysteresis to the BOD33 threshold level.

The hysteresis functionality can be used in both Continuous and Sampling Mode.

13.11.6.5.8 Sleep Mode Operation

Standby Mode

The BOD33 can be used in standby mode if the BOD is enabled and the corresponding Run in Standby bit is written to '1' (BOD33.RUNSTDBY).

The BOD33 can be configured to work in either Continuous or Sampling Mode by writing a '1' to the Configuration in Standby Sleep Mode bit (BOD33.STDBYCFG).

Backup Mode

In Backup mode, the BOD12 is automatically disabled.

If the BOD33 is enabled and the Run in Backup sleep mode bit in the BOD33 register (BOD33.RUNBKUP) is written to '1', the BOD33 will operate in Sampling mode. In this state, the voltage monitored by BOD33 is always the supply of the backup domain, i.e. VDD or VBAT.

13.11.6.6 Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Automatic Power Switch Ready Ready (APSWRDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

11.2 Nested Vector Interrupt Controller

13.11.6.7 Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD33 Enable bit (BOD33.ENABLE) need synchronization when written.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD33 Control register. The Synchronization Ready bit (STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY is '0') will generate an error without stalling the APB bus.

13.11.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8						VCORERDY	APWSRDY	VREGRDY
		23:16								
		31:24								
0x04	INTENSET	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8						VCORERDY	APWSRDY	VREGRDY
		23:16								
		31:24								
0x08	INTFLAG	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8						VCORERDY	APWSRDY	VREGRDY
		23:16								
		31:24								
0x0C	STATUS	7:0						B33SRDY	BOD33DET	BOD33RDY
		15:8					BBPS	VCORERDY	APWSRDY	VREGRDY
		23:16								
		31:24								
0x10	BOD33	7:0	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
		15:8	PSEL[3:0]					VMON		ACTCFG
		23:16			LEVEL[5:0]					
		31:24			BKUPLEVEL[5:0]					
0x14 ... 0x17	Reserved									
0x18	VREG	7:0		RUNSTDBY	STDBYPL0			SEL	ENABLE	
		15:8								LPEFF
		23:16					VSVSTEP[3:0]			
		31:24	VSPER[7:0]							
0x1C	VREF	7:0	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
		15:8								
		23:16					SEL[3:0]			
		31:24								
0x20	BBPS	7:0					PSOKEN	WAKEEN	CONF[1:0]	
		15:8								
		23:16								
		31:24								
0x24	BKOUT	7:0							EN[1:0]	
		15:8							CLR[1:0]	
		23:16							SET[1:0]	
		31:24							RTCTGL[1:0]	
0x28	BKIN	7:0						BKIN[2:0]		
		15:8								
		23:16								
		31:24								

13.11.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). PAC Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to 13.11.5.8 Register Access Protection for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. Refer to 13.11.6.7 Synchronization for details.

13.11.8.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the VDDCORE Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Automatic Power Switch Ready Interrupt Enable bit, which disables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the APWSRDY Interrupt Flag is set.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

13.11.8.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the VDDCORE Ready Interrupt Enable bit, which enables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Automatic Power Switch Ready Interrupt Enable bit, which enables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the Automatic Power Switch Ready Interrupt Flag is set.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

13.11.8.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x0000010X - X= determined from NVM User Row (0xX=0bx00y)
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R/W	R/W	R/W
Reset						0	0	y

Bit 10 – VCORERDY VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.
This flag is set on a zero-to-one transition of the VDDCORE Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VCORERDY interrupt flag.

Bit 9 – APWSRDY Automatic Power Switch Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) and will generate an interrupt request if INTENSET.APWSRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the APWSRDY interrupt flag.

Bit 8 – VREGRDY Voltage Regulator Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VREGRDY interrupt flag.

Bit 2 – B33SRDY BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

Bit 1 – BOD33DET BOD33 Detection
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

Bit 0 – BOD33RDY BOD33 Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Ready interrupt flag.
The BOD33 can be enabled.

Related Links

10.3 NVM User Row Mapping

13.11.8.4 Status

Name: STATUS
Offset: 0x0C
Reset: Determined from NVM User Row
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					BBPS	VCORERDY	APWSRDY	VREGRDY
Access					R	R	R	R
Reset					0	1	0	1

Bit	7	6	5	4	3	2	1	0
						B33SRDY	BOD33DET	BOD33RDY
Access						R	R	R
Reset						0	0	y

Bit 11 – BBPS Battery Backup Power Switch

Value	Description
0	the backup domain is supplied by VDD.
1	the backup domain is supplied by VBAT.

Bit 10 – VCORERDY VDDCORE Voltage Ready

Value	Description
0	the VDDCORE voltage is not as expected.
1	the VDDCORE voltage is the target voltage.

Bit 9 – APWSRDY Automatic Power Switch Ready

Value	Description
0	The Automatic Power Switch is not ready.
1	The Automatic Power Switch is ready.

Bit 8 – VREGRDY Voltage Regulator Ready

Value	Description
0	The selected voltage regulator in VREG.SEL is not ready.
1	The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator.

Bit 2 – B33SRDY BOD33 Synchronization Ready

Value	Description
0	BOD33 synchronization is ongoing.
1	BOD33 synchronization is complete.

Bit 1 – BOD33DET BOD33 Detection

Value	Description
0	No BOD33 detection.

Value	Description
1	BOD33 has detected that the I/O power supply is going below the BOD33 reference value.

Bit 0 – BOD33RDY BOD33 Ready
The BOD33 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD33 is not ready.
1	BOD33 is ready.

Related Links
10.3 NVM User Row Mapping

13.11.8.5 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10
Reset: Determined from NVM User Row
Property: Write-Synchronized, Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			BKUPLEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]					VMON		ACTCFG
Access	R/W	R/W	R/W	R/W		R/W		R/W
Reset	0	0	0	0		0		0

Bit	7	6	5	4	3	2	1	0
	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	y	y	0	z	

Bits 29:24 – BKUPLEVEL[5:0] BOD33 Threshold Level on VBAT or in Backup Sleep Mode
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT or in backup sleep mode.
This bit field is not synchronized.

Bits 21:16 – LEVEL[5:0] BOD33 Threshold Level on VDD
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors VDD except in backup sleep mode.
These bits are loaded from NVM User Row at start-up.
This bit field is not synchronized.

Bits 15:12 – PSEL[3:0] Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode. The input clock comes from the OSCULP32K 1KHz output.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384

Value	Name	Description
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 10 – VMON Voltage Monitored in Active and Standby Mode
This bit is not synchronized.

Value	Description
0	The BOD33 monitors the VDD power pin in active and standby mode.
1	The BOD33 monitors the VBAT power pin in active and standby mode.

Bit 8 – ACTCFG BOD33 Configuration in Active Sleep Mode
This bit is not synchronized.

Value	Description
0	In active mode, the BOD33 operates in continuous mode.
1	In active mode, the BOD33 operates in sampling mode.

Bit 7 – RUNBKUP BOD33 Configuration in Backup Sleep Mode
This bit is not synchronized.

Value	Description
0	In backup sleep mode, the BOD33 is disabled.
1	In backup sleep mode, the BOD33 is enabled and configured in sampling mode.

Bit 6 – RUNSTDBY Run in Standby
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is disabled.
1	In standby sleep mode, the BOD33 is enabled.

Bit 5 – STDBYCFG BOD33 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD33 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0] BOD33 Action
These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.
These bits are loaded from NVM User Row at start-up.
This bit field is not synchronized.

Value	Name	Description
0x0	NONE	No action
0x1	RESET	The BOD33 generates a reset
0x2	INT	The BOD33 generates an interrupt
0x3	BKUP	The BOD33 puts the device in backup sleep mode if VMON=0. No action if VMON=1.

Bit 2 – HYST Hysteresis
This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.
This bit is loaded from NVM User Row at start-up.
This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE Enable
This bit is loaded from NVM User Row at start-up.
This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

Related Links

15. Electrical Characteristics

10.3 NVM User Row Mapping

13.11.8.6 Voltage Regulator System (VREG) Control

Name: VREG
Offset: 0x18
Reset: 0x00000002
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VSPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VSVSTEP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								LPEFF
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYPL0			SEL	ENABLE	
Access		R/W	R/W			R/W	R/W	
Reset		0	1			0	1	

Bits 31:24 – VSPER[7:0] Voltage Scaling Period
This bitfield sets the period between the voltage steps when the VDDCORE voltage is changing in µs.
If VSPER=0, the period between two voltage steps is 1µs.

Bits 19:16 – VSVSTEP[3:0] Voltage Scaling Voltage Step
This field sets the voltage step height when the VDDCORE voltage is changing to reach the target VDDCORE voltage.
The voltage step is equal to $2^{VSVSTEP*}$ min_step.
See the Electrical Characteristics chapter for the min_step voltage level.

Bit 8 – LPEFF Low power Mode Efficiency

Value	Description
0	The voltage regulator in Low power mode has the default efficiency and supports the whole VDD range (1.62V to 3.6V).
1	The voltage regulator in Low power mode has the highest efficiency and supports a limited VDD range (2.5V to 3.6V).

Bit 6 – RUNSTDBY Run in Standby

Value	Description
0	The voltage regulator is in low power mode in Standby sleep mode.
1	The voltage regulator is in normal mode in Standby sleep mode.

Bit 5 – STDBYPL0 Standby in PL0
This bit selects the performance level (PL) of the main voltage regulator for the Standby sleep mode. This bit is only considered when RUNSTDBY=1.

Value	Description
0	In Standby sleep mode, the voltage regulator remains in the current performance level.
1	In Standby sleep mode, the voltage regulator is used in PL0.

Bit 2 – SEL Voltage Regulator Selection
This bit is loaded from NVM User Row at start-up.

Value	Description
0	The voltage regulator in active mode is a LDO voltage regulator.
1	The voltage regulator in active mode is a buck converter.

Bit 1 – ENABLE Enable

Value	Description
0	The voltage regulator is disabled.
1	The voltage regulator is enabled.

13.11.8.7 Voltage References System (VREF) Control

Name: VREF
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
Access	R/W	R/W				R/W	R/W	
Reset	0	0				0	0	

Bits 19:16 – SEL[3:0] Voltage Reference Selection
These bits select the Voltage Reference for the ADC/SDADC.

Value	Description
0x0	1.0V voltage reference typical value
0x1	1.1V voltage reference typical value
0x2	1.2V voltage reference typical value
0x3	1.25V voltage reference typical value
0x4	2.0V voltage reference typical value
0x5	2.2V voltage reference typical value
0x6	2.4V voltage reference typical value
0x7	2.5V voltage reference typical value
Others	Reserved

Note:

1. 2.0V to 2.5V can be used when the supply voltage (VDDANA) is higher than 2.7V.

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows for enabling or disabling the voltage reference depending on the peripheral requests.

Value	Description
0	The voltage reference is always on, if enabled.
1	The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run In Standby
The bit controls how the voltage reference behaves during the standby sleep mode.

Value	Description
0	The voltage reference is halted during standby sleep mode.

Value	Description
1	The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND = 1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND = 0, the voltage reference will always be running in standby sleep mode.

Bit 2 – VREFOE Voltage Reference Output Enable

Value	Description
0	The Voltage Reference output is not available as an ADC input channel.
1	The Voltage Reference output is routed to an ADC input channel.

Bit 1 – TSEN Temperature Sensor Enable

Value	Description
0	Temperature Sensor is disabled.
1	Temperature Sensor is enabled and routed to an ADC input channel.

13.11.8.8 Battery Backup Power Switch (BBPS) Control

Name: BBPS
Offset: 0x20
Reset: 0x0000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PSOKEN	WAKEEN	CONF[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – PSOKEN Power Supply OK Enable

Value	Description
0	The PSOK pin is not used.
1	The PSOK pin is used to determine the status of the Main Power Supply.

Bit 2 – WAKEEN Wake Enable

Value	Description
0	The device is not woken up when switched from battery backup power to Main Power.
1	The device is woken up when switched from battery backup power to Main Power.

Bits 1:0 – CONF[1:0] Battery Backup Power Switch Configuration

Value	Name	Description
0x0	NONE	The backup domain is always supplied by Main Power.
0x1	APWS	The power switch is handled by the Automatic Power Switch.
0x2	FORCED	The backup domain is always supplied by Battery Backup Power.
0x3	BOD33	The power switch is handled by the BOD33.

13.11.8.9 Backup Output (BKOUT) Control

Name: BKOUT
Offset: 0x24
Reset: 0x0000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
							RTCTGL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							SET[1:0]	
Access							W	W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							CLR[1:0]	
Access							W	W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							EN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 25:24 – RTCTGL[1:0] RTC Toggle Output

Value	Description
0	The output will not toggle on RTC event.
1	The output will toggle on RTC event.

Bits 17:16 – SET[1:0] Set Output
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will set the corresponding output.
Reading this bit returns '0'.

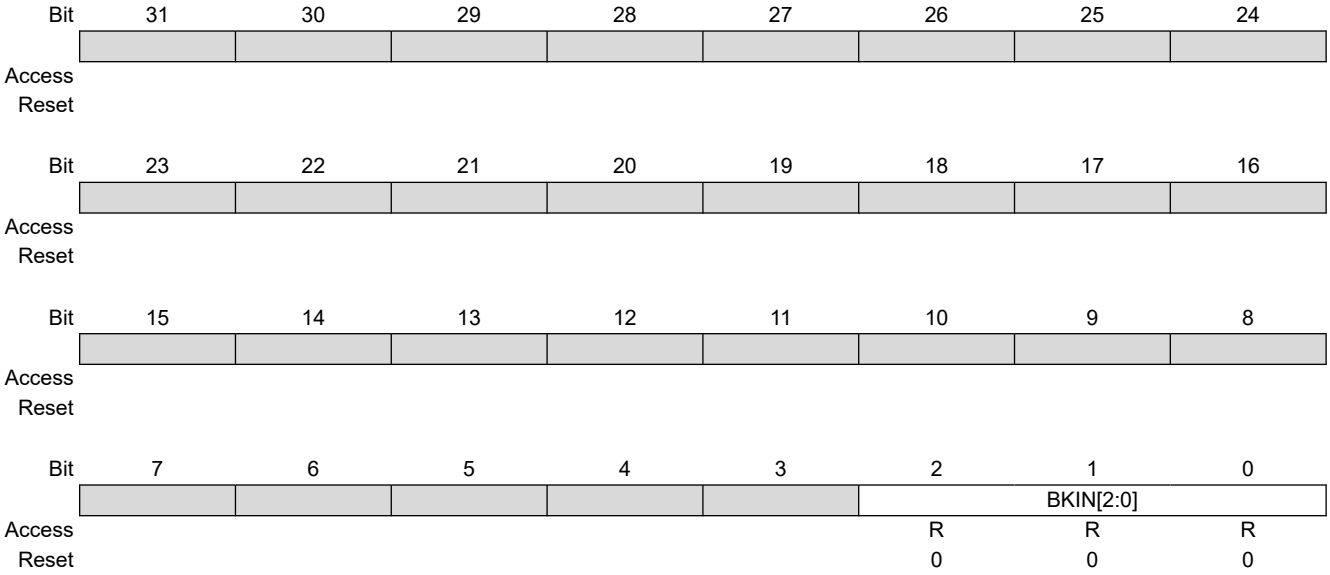
Bits 9:8 – CLR[1:0] Clear Output
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding output.
Reading this bit returns '0'.

Bits 1:0 – EN[1:0] Enable Output

Value	Description
0	The output is not enabled.
1	The output is enabled and driven by the SUPC.

13.11.8.10 Backup Input (BKIN) Value

Name: BKIN
Offset: 0x28
Reset: 0x0000
Property: -



Bits 2:0 – BKIN[2:0] Backup I/O Data Input Value
These bits are cleared when the corresponding backup I/O pin detects a logical low level on the input pin or when the backup I/O is not enabled.
These bits are set when the corresponding backup I/O pin detects a logical high level on the input pin when the backup I/O is enabled.

BKIN[2:0]	PAD	Description
BKIN[0]	PSOK	If BBPS.PSOKEN=1, BKIN[0] will give the input value of the PSOK pin
BKIN[1]	OUT[0]	If BKOUT.EN[0]=1, BKIN[1] will give the input value of the OUT[0] pin
BKIN[2]	OUT[1]	If BKOUT.EN[1]=1, BKIN[2] will give the input value of the OUT[1] pin

13.12 WDT – Watchdog Timer

13.12.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

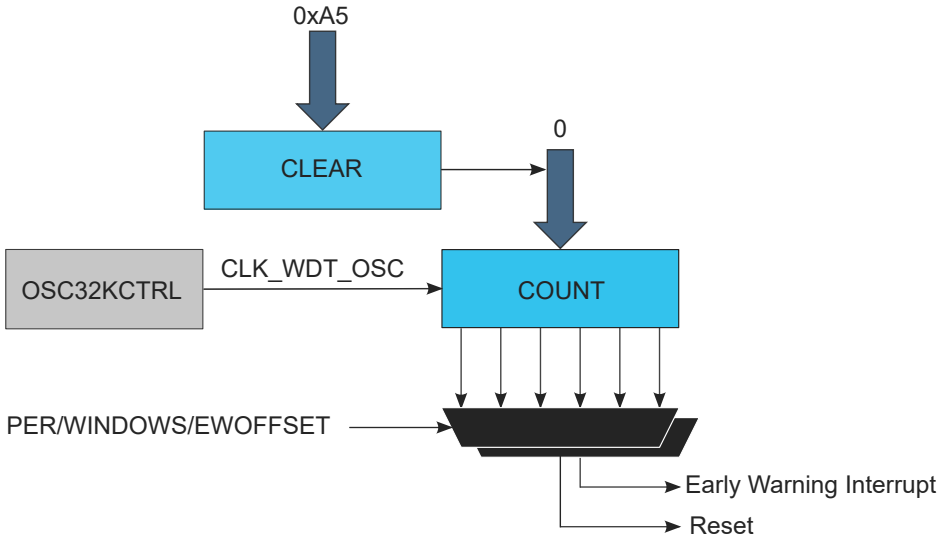
When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

13.12.2 Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

13.12.3 Block Diagram

Figure 13-40. WDT Block Diagram



13.12.4 Signal Description

Not applicable.

13.12.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.12.5.1 I/O Lines

Not applicable.

13.12.5.2 Power Management

The WDT can continue to operate in any sleep mode where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.12.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1.024 kHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

The CLK_WDT_OSC CLOCK is sourced from the clock of the internal Ultra Low-Power Oscillator (OSCULP32K). Due to ultra low-power design, the oscillator is not very accurate; therefore, the exact time-out period may vary from

device-to-device. This variation must be considered when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 13.12.6.7 Synchronization for further details.

13.12.5.4 DMA

Not applicable.

13.12.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the WDT interrupt(s) requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

13.12.5.6 Events

Not applicable.

13.12.5.7 Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

13.12.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.12.5.9 Analog Connections

Not applicable.

13.12.6 Functional Description

13.12.6.1 Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 13-38. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

13.12.6.2 Basic Operation

13.12.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

13.12.6.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

Related Links

10.3 NVM User Row Mapping

13.12.6.2.3 Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

13.12.6.2.4 Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

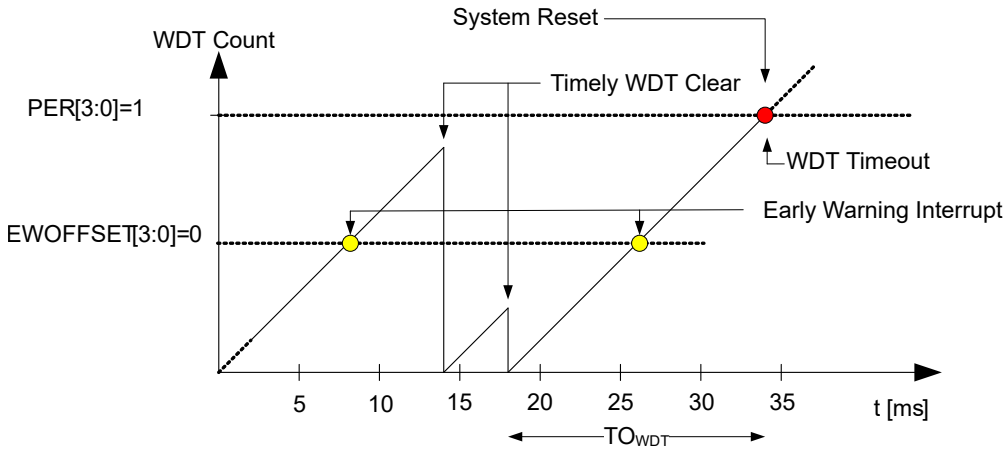
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 13-41. Normal-Mode Operation



13.12.6.2.5 Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

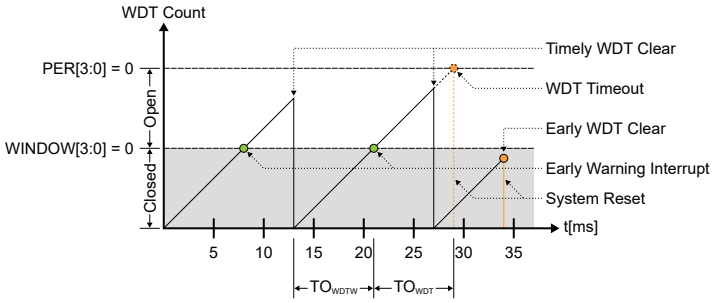
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW}. The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 13-42. Window-Mode Operation



13.12.6.3 DMA Operation

Not applicable.

13.12.6.4 Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the 13.12.8.6 INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

13.8 PM – Power Manager

13.12.6.5 Events

Not applicable.

13.12.6.6 Sleep Mode Operation

The WDT will continue to operate in any sleep mode where the source clock is active except backup mode. The WDT interrupts can be used to wake up the device from a sleep mode. An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

Related Links

13.12.8.1 CTRLA

13.12.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following registers are synchronized when written:

- Enable bit in Control A register (CTRLA.ENABLE)
- Window Enable bit in Control A register (CTRLA.WEN)
- Always-On bit in control Control A (CTRLA.ALWAYSON)
- Watchdog Clear register (CLEAR)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

13.12.6.8 Additional Features

13.12.6.8.1 Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 13-39. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

13.12.6.8.2 Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

13.12.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ALWAYSON					WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0					EWOFFSET[3:0]			
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	7:0				CLEAR	ALWAYSON	WEN	ENABLE	
		15:8								
		23:16								
		31:24								
0x0C	CLEAR	7:0	CLEAR[7:0]							

13.12.8 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.12.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.12.6.7 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.12.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: N/A - Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ALWAYSON					WEN	ENABLE	
Access	R/W					R/W	R/W	
Reset	0					0	0	

Bit 7 – ALWAYSON Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.
Writing a '0' to this bit has no effect.
This bit is not Enable-Protected.
This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 2 – WEN Watchdog Timer Window Mode Enable
This bit enables Window mode. It can only be written if the peripheral is disabled unless CTRLA.ALWAYSON=1. The initial value of this bit is loaded from Flash Calibration.
This bit is loaded from NVM User Row at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Bit 1 – ENABLE Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.
Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not Enable-Protected.
This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

Related Links
10.3 NVM User Row Mapping

13.12.8.2 Configuration

Name: CONFIG
Offset: 0x01
Reset: Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – WINDOW[3:0] Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1.024kHz CLK_WDT_OSC clock.
These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

Bits 3:0 – PER[3:0] Time-Out Period
These bits determine the watchdog time-out period as a number of 1.024kHz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.
These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

Related Links
10.3 NVM User Row Mapping

13.12.8.3 Early Warning Control

Name: EWCTRL
Offset: 0x02
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					EWOFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					x	x	x	x

Bits 3:0 – EWOFFSET[3:0] Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at start-up. Refer to 10.3 NVM User Row Mapping.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	—	Reserved

13.12.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

13.12.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

13.12.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: N/A

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW Early Warning
This flag is cleared by writing a '1' to it.
This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning interrupt flag.

13.12.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CLEAR	ALWAYSON	WEN	ENABLE	
Access				R	R	R	R	
Reset				0	0	0	0	

Bit 4 – CLEAR Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 3 – ALWAYSON Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 2 – WEN Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

13.12.8.8 Clear

Name: CLEAR
Offset: 0x0C
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CLEAR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CLEAR[7:0] Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.
In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.
In both modes, writing any other value than 0xA5 will issue an immediate system Reset.

13.13 RTC – Real-Time Counter

13.13.1 Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

13.13.2 Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- One 32-bit or two 16-bit compare values
- Clock/Calendar mode
  - Time in seconds, minutes and hours (12/24)
  - Date in day of month, month and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- 2 general purpose registers

13.13.3 Block Diagram

Figure 13-43. RTC Block Diagram (Mode 0 — 32-Bit Counter)

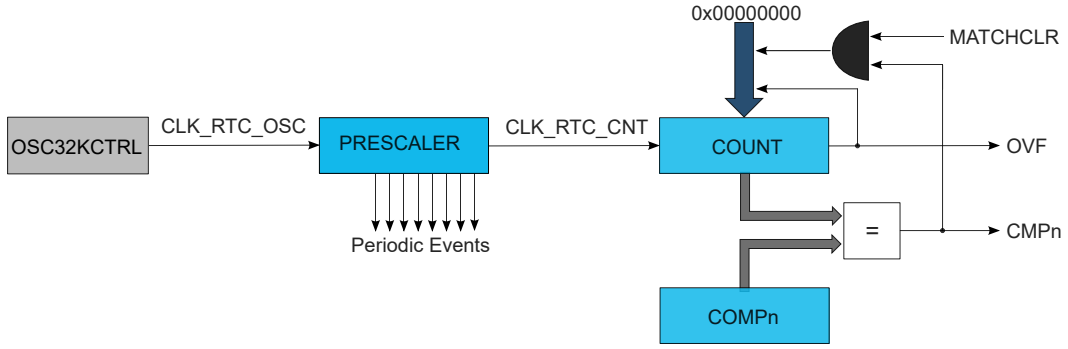


Figure 13-44. RTC Block Diagram (Mode 1 — 16-Bit Counter)

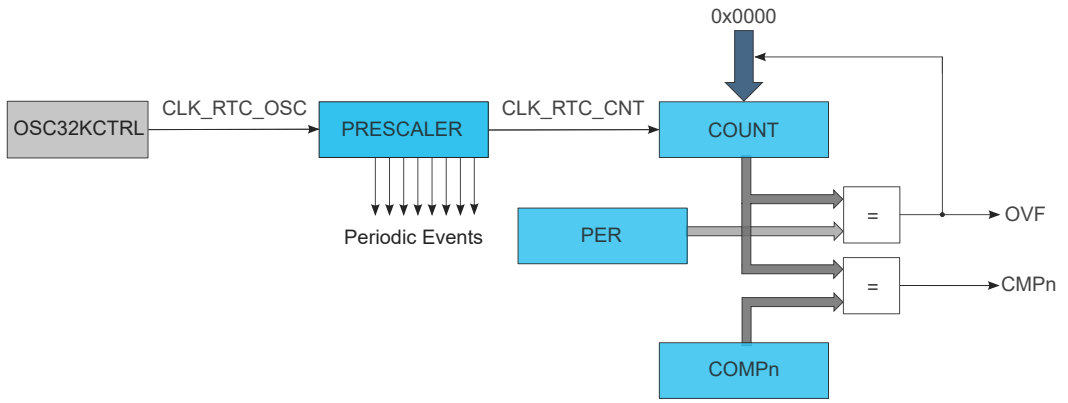
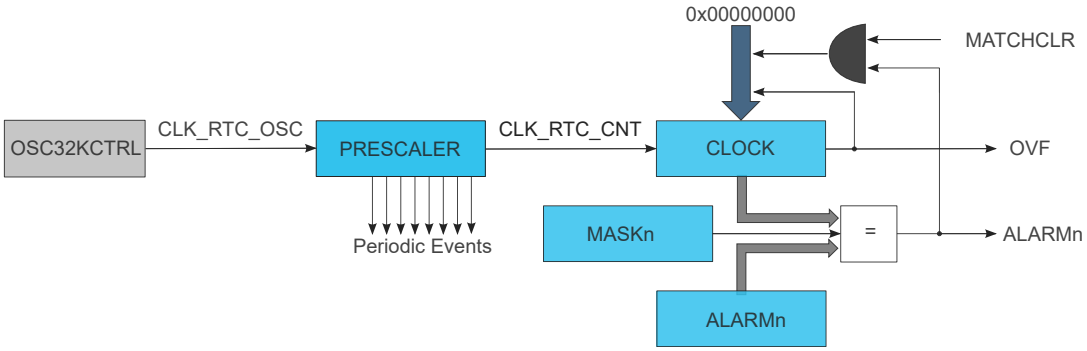


Figure 13-45. RTC Block Diagram (Mode 2 — Clock/Calendar)



Related Links

13.13.6.2.3 32-Bit Counter (Mode 0)
13.13.6.2.4 16-Bit Counter (Mode 1)
13.13.6.2.5 Clock/Calendar (Mode 2)

13.13.4 Signal Description

Not applicable.

Related Links

7. I/O Multiplexing and Considerations

13.13.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.13.5.1 I/O Lines

Not applicable.

13.13.5.2 Power Management

The RTC will continue to operate in any sleep mode where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

The RTC will be reset only at power-on (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

Related Links

13.8 PM – Power Manager

13.13.5.3 Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in Peripheral Clock Masking section.

A 32KHz or 1KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32KHz oscillator controller (OSC32KCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 13.13.6.7 Synchronization for further details.

Related Links

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.6.6.2.6 Peripheral Clock Masking

13.13.5.4 DMA

Not applicable.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.13.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.13.5.6 Events

The events are connected to the *Event System*.

Related Links

13.18 EVSYS – Event System

13.13.5.7 Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to 13.13.8.6 DBGCTRL for details.

13.13.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register
- General Purpose (GPx) registers

Write-protection is denoted by the "PAC Write-Protection" property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller for details.

13.13.5.9 Analog Connections

A 32.768kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended crystal characteristics and load capacitors.

Related Links

15. Electrical Characteristics

13.13.6 Functional Description

13.13.6.1 Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

13.13.6.2 Basic Operation

13.13.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following register is enable-protected

- Event Control register (EVCTRL)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{\mathrm{CLK\_RTC\_CNT}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{\mathrm{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{\mathrm{CLK\_RTC\_OSC}}$, and $f_{\mathrm{CLK\_RTC\_CNT}}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

13.13.6.2.2 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

13.13.6.2.3 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in Figure 13-43. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP0). When a compare match occurs, the Compare 0 Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMP0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMP0 and INTFLAG.OVF will both be set simultaneously on a compare match with COMP0.

13.13.6.2.4 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode as shown in Figure 13-44. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

13.13.6.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in Figure 13-45. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm 0 Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm 0 Mask register (MASK0.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see 13.13.6.8.1 Periodic Intervals).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0.

13.13.6.3 DMA Operation

Not applicable.

13.13.6.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to 13.13.6.8.1 Periodic Intervals for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

Related Links
11.2 Nested Vector Interrupt Controller

13.13.6.5 Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to 13.13.6.8.1 Periodic Intervals for details.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

Related Links
13.18 EVSYS – Event System

13.13.6.6 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

13.13.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

13.4.3 Register Synchronization

13.13.6.8 Additional Features

13.13.6.8.1 Periodic Intervals

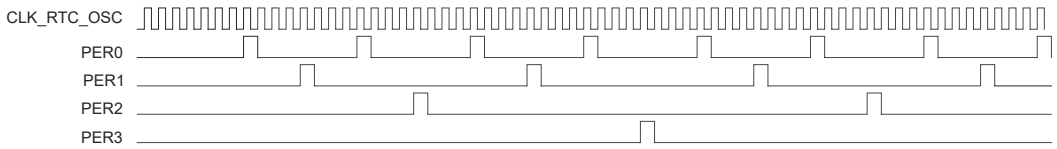
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{\mathrm{PERIODIC(n)}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{n+3}}$$

$f_{CLK\_RTC\_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 13-46. Example Periodic Events



13.13.6.8.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 4096 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 240 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{4096 \cdot 240} \cdot 10^6 \text{ppm}$$

This results in a resolution of 1.017ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

13.13.6.8.3 General Purpose Registers

The RTC includes two General Purpose registers (GPn). These registers are reset only when the RTC is reset, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

13.13.7 Register Summary - COUNT32

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO							CMPEO0
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16							GP1	GP0
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
0x20	COMP	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
		23:16	COMP[23:16]							
		31:24	COMP[31:24]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

13.13.8 Register Description - COUNT32

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.13.8.1 Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit defines if the counter is cleared or not on a match.
This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0] Operating Mode
This bit group defines the operating mode of the RTC.
This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter

Value	Name	Description
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

13.13.8.2 Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO							CMPEO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 8 – CMPEO0 Compare 0 Event Output Enable

Value	Description
0	Compare 0 event is disabled and will not be generated.
1	Compare 0 event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

13.13.8.3 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – CMP0 Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Compare 0 Interrupt Enable bit, which disables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.8.4 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – CMP0 Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Compare 0 Interrupt Enable bit, which enables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.8.5 Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 8 – CMP0 Compare 0
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMP0 is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare 0 interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

13.13.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

13.13.8.7 Synchronization Busy in COUNT32 mode (CTRLA.MODE = 0)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: —

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GP1	GP0
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 16, 17 – GPn General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for the GPn register is complete.
1	Write synchronization for the GPn register is ongoing.

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for the CTRLA.COUNTSYNC bit is ongoing.

Bit 5 – COMP0 Compare 0 Synchronization Busy Status

Value	Description
0	Write synchronization for the COMP0 register is complete.
1	Write synchronization for the COMP0 register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for the COUNT register is complete.
1	Read/write synchronization for the COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for the FREQCORR register is complete.
1	Read/write synchronization for the FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for the CTRLA.ENABLE bit is complete.
1	Read/write synchronization for the CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for the CTRLA.SWRST bit is complete.
1	Read/write synchronization for the CTRLA.SWRST bit is ongoing.

13.13.8.8 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

13.13.8.9 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value
These bits define the value of the 32-bit RTC counter in mode 0.

13.13.8.10 Compare 0 Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMP
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0] Compare Value
The 32-bit value of COMPn is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is one.

13.13.8.11 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 13.13.6.8.3 General Purpose Registers.

13.13.9 Register Summary - COUNT16

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
		15:8	COUNTSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO						CMPEO1	CMPEO0
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF						CMP1	CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF						CMP1	CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF						CMP1	CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
		15:8	COUNTSYNC							
		23:16							GPn[1:0]	
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x1A ... 0x1B	Reserved									
0x1C	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x1E ... 0x1F	Reserved									
0x20	COMP	7:0	COMP[7:0]							
		15:8	COMP[15:8]							
0x22 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

13.13.10 Register Description - COUNT16

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.13.10.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0] Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

13.13.10.2 Event Control in COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO						CMPEO1	CMPEO0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bits 8, 9 – CMPEOn Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated. [n = 7..0]
1	Periodic Interval n event is enabled and will be generated. [n = 7..0]

13.13.10.3 Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 8, 9 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.10.4 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 8, 9 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.10.5 Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 8, 9 – CMPn Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

13.13.10.6 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

13.13.10.7 Synchronization Busy in COUNT16 mode (CTRLA.MODE = 1)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: —

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn[1:0] General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for the GPn register is complete.
1	Write synchronization for the GPn register is ongoing.

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for the CTRLA.COUNTSYNC bit is ongoing.

Bit 5 – COMP0 Compare 0 Synchronization Busy Status

Value	Description
0	Write synchronization for the COMP0 register is complete.
1	Write synchronization for the COMP0 register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for the COUNT register is complete.
1	Read/write synchronization for the COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for the FREQCORR register is complete.
1	Write synchronization for the FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.ENABLE bit is complete.
1	Write synchronization for the CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.SWRST bit is complete.
1	Write synchronization for the CTRLA.SWRST bit is ongoing.

13.13.10.8 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

13.13.10.9 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value
These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

13.13.10.10 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0] Counter Period
These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

13.13.10.11 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: 0x20
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0] Compare Value
The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.

13.13.10.12 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 13.13.6.8.3 General Purpose Registers.

13.13.11 Register Summary - CLOCK

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
		15:8	CLOCKSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
		15:8	OVFEO							ALARMO0
		23:16								
		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							ALARM0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							ALARM0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
		15:8	OVF							ALARM0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
		15:8	CLOCKSYNC				MASK0			
		23:16							GPn[1:0]	
		31:24								
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]		SECOND[5:0]					
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4]
		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM	7:0	MINUTE[1:0]		SECOND[5:0]					
		15:8	HOUR[3:0]				MINUTE[5:2]			
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4]
		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	MASK	7:0						SEL[2:0]		
0x25 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
		15:8	GP[15:8]							
		23:16	GP[23:16]							
		31:24	GP[31:24]							

13.13.12 Register Description - CLOCK

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.13.12.1 Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour

Value	Description
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

13.13.12.2 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO							ALARMO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 8 – ALARMO0 Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

13.13.12.3 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – ALARM0 Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm 0 Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.12.4 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – ALARM0 Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm 0 Interrupt Enable bit, which enables the Alarm 0 interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

13.13.12.5 Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 8 – ALARM0 Alarm 0
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARM0 is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm 0 interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

13.13.12.6 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

13.13.12.7 Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPn[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				MASK0			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 17:16 – GPn[1:0] General Purpose n Synchronization Busy Status

Value	Description
0	Write synchronization for GPn register is complete.
1	Write synchronization for GPn register is ongoing.

Bit 15 – CLOCKSYNC Clock Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bit 11 – MASK0 Mask 0 Synchronization Busy Status

Value	Description
0	Write synchronization for MASK0 register is complete.
1	Write synchronization for MASK0 register is ongoing.

Bit 5 – ALARM0 Alarm 0 Synchronization Busy Status

Value	Description
0	Write synchronization for ALARM0 register is complete.
1	Write synchronization for ALARM0 register is ongoing.

Bit 3 – CLOCK Clock Register Synchronization Busy Status

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

13.13.12.8 Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

13.13.12.9 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The year offset with respect to the reference year (defined in software).
The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0] Month
1 – January
2 – February
...
12 – December

Bits 21:17 – DAY[4:0] Day
Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0] Hour
When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0] Minute
0 – 59

Bits 5:0 – SECOND[5:0] Second
0 – 59

13.13.12.10 Alarm Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: ALARM
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

The 32-bit value of ALARM is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year
The alarm year. Years are only matched if MASK.SEL is 6

Bits 25:22 – MONTH[3:0] Month
The alarm month. Months are matched only if MASK.SEL is greater than 4.

Bits 21:17 – DAY[4:0] Day
The alarm day. Days are matched only if MASK.SEL is greater than 3.

Bits 16:12 – HOUR[4:0] Hour
The alarm hour. Hours are matched only if MASK.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0] Minute
The alarm minute. Minutes are matched only if MASK.SEL is greater than 1.

Bits 5:0 – SECOND[5:0] Second
The alarm second. Seconds are matched only if MASK.SEL is greater than 0.

13.13.12.11 Alarm Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEL[2:0] Alarm Mask Selection
These bits define which bit groups of ALARM are valid.

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years
0x7	-	Reserved

13.13.12.12 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see 13.13.6.8.3 General Purpose Registers.

13.14 DMAC – Direct Memory Access Controller

13.14.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the low-power (LP) SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to LP SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.

- The *write-back bus* is used to write the transfer descriptor back to LP SRAM.

All buses are AHB master interfaces but the AHB/APB Bridge bus, which is an APB slave interface.

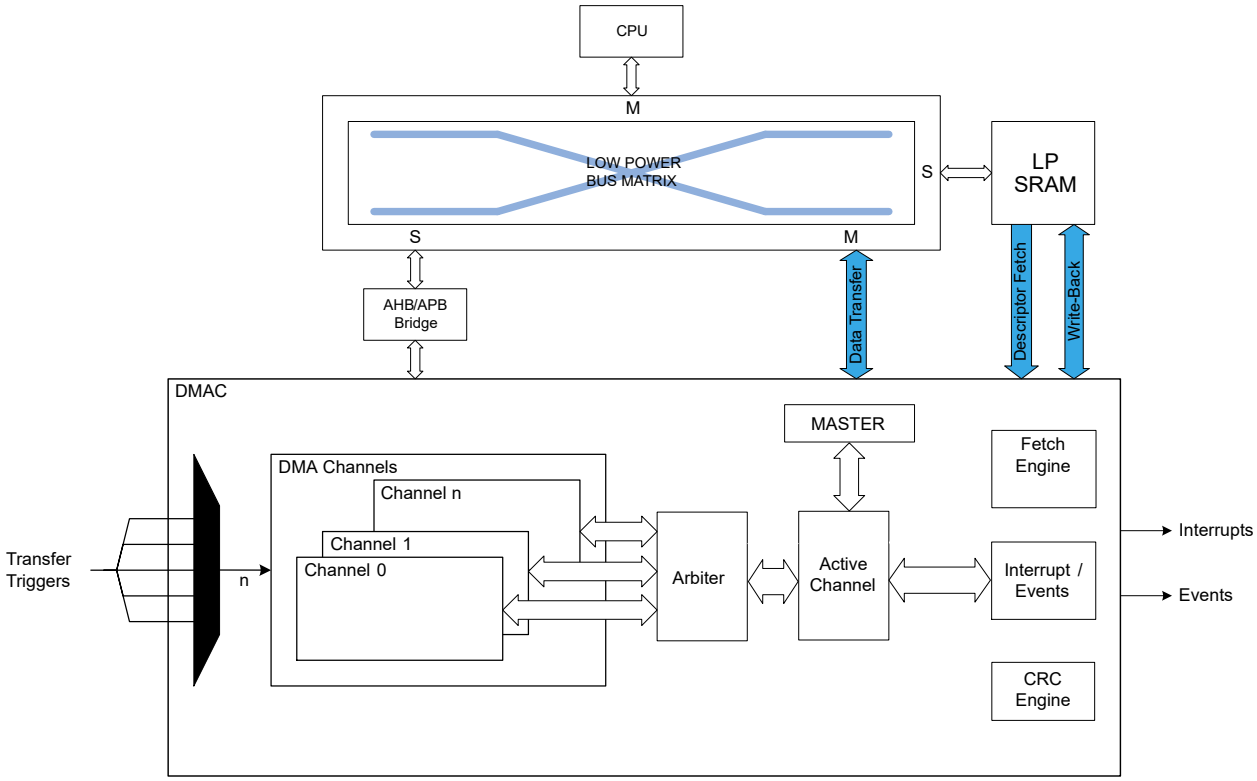
The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

13.14.2 Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16channels
  - Enable 16 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
  - Static
  - Configurable increment scheme
- Optional interrupt generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- 4 event inputs
  - One event input for each of the 4 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 4 event outputs
  - One output event for each of the 4 least significant DMA channels
  - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)

13.14.3 Block Diagram

Figure 13-47. DMAC Block Diagram



13.14.4 Signal Description

Not applicable.

13.14.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.14.5.1 I/O Lines

Not applicable.

13.14.5.2 Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

Related Links

13.8 PM – Power Manager

13.14.5.3 Clocks

The DMAC bus clock (CLK_DMAC_APB) must be configured and enabled in the Main Clock module before using the DMAC.

This bus clock (CLK_DMAC_APB) is always synchronous to the module clock (CLK_DMAC_AHB), but can be divided by a prescaler and may run even when the module clock is turned off.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.14.5.4 DMA

Not applicable.

13.14.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.14.5.6 Events

The events are connected to the event system.

Related Links

13.18 EVSYS – Event System

13.14.5.7 Debug Operation

When the CPU is halted in debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to 13.14.8.6 DBGCTRL for details.

13.14.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.14.5.9 Analog Connections

Not applicable.

13.14.6 Functional Description

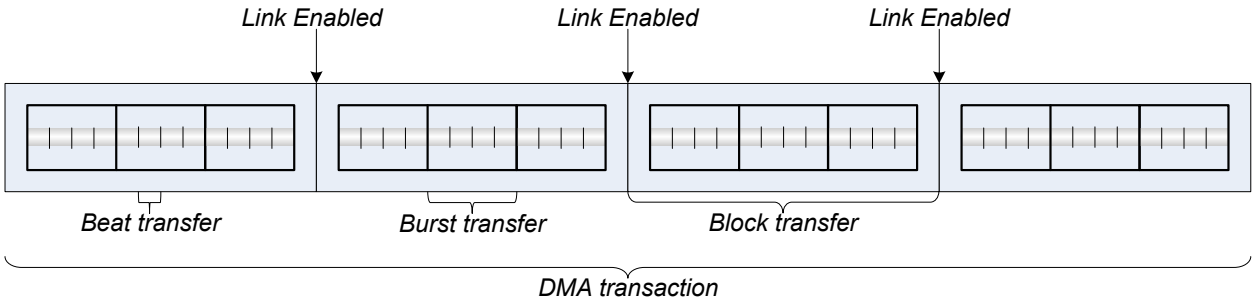
13.14.6.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

13.14.6.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

Figure 13-48. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.

- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in Low-Power (LP) SRAM. For further details on the transfer descriptor refer to 13.14.6.2.3 Transfer Descriptors.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to 13.14.6.3.1 Linked Descriptors.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

13.14.6.1.2 CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to 13.14.6.3.7 CRC Operation for details.

13.14.6.2 Basic Operation

13.14.6.2.1 Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CTRL.CRCENABLE=0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE=0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE=0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The LP SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The LP SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register

- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)
- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

13.14.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (13.14.8.1 CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to 13.14.8.1 CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (13.14.8.1 CTRL.CRCENABLE). The CRC is disabled by writing a '0' to 13.14.8.1 CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (13.14.8.1 CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

13.14.6.2.3 Transfer Descriptors

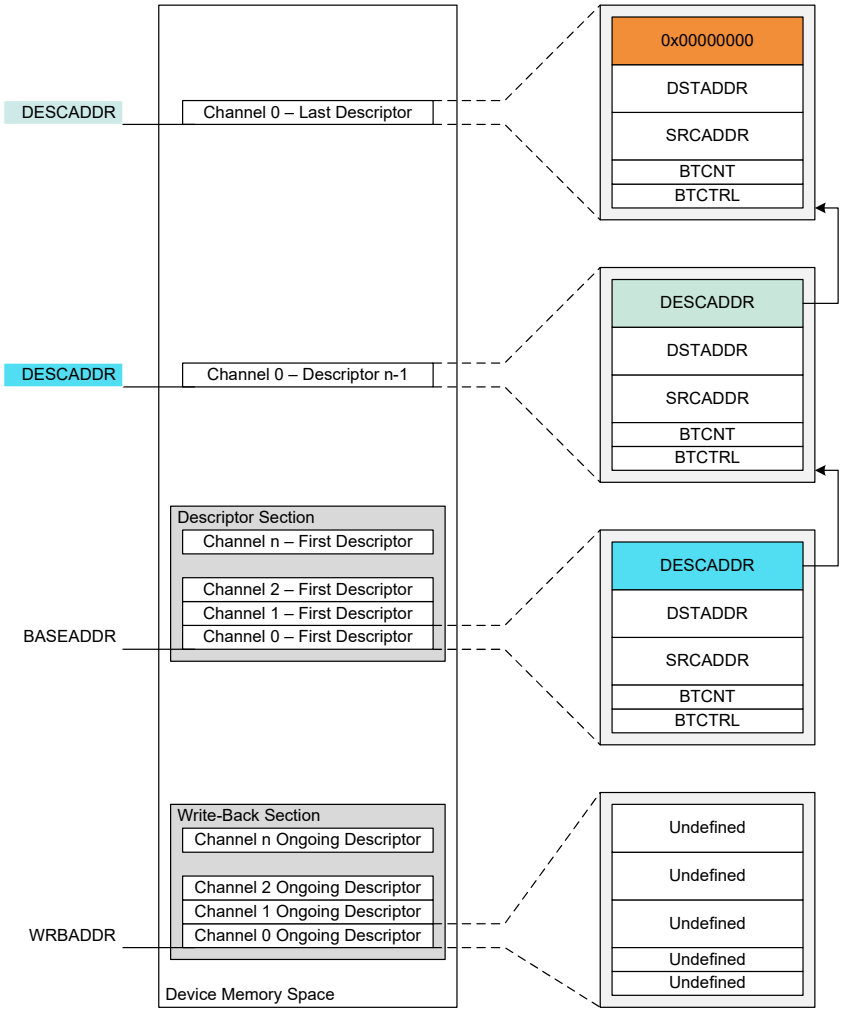
Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

All transfer descriptors must reside in LP SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to 13.14.6.3.1 Linked Descriptors.

The write-back memory section is the section where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel 0. All ongoing transfer descriptors will be stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For further details on linked descriptors, refer to 13.14.6.3.1 Linked Descriptors.

Figure 13-49. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to always use the less significant DMA channels if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less LP SRAM. In addition, the latency from fetching the first descriptor of a transaction to the first burst transfer is executed, is reduced.

13.14.6.2.4 Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request, it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The active channel is the DMA channel being granted access to perform its next burst transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

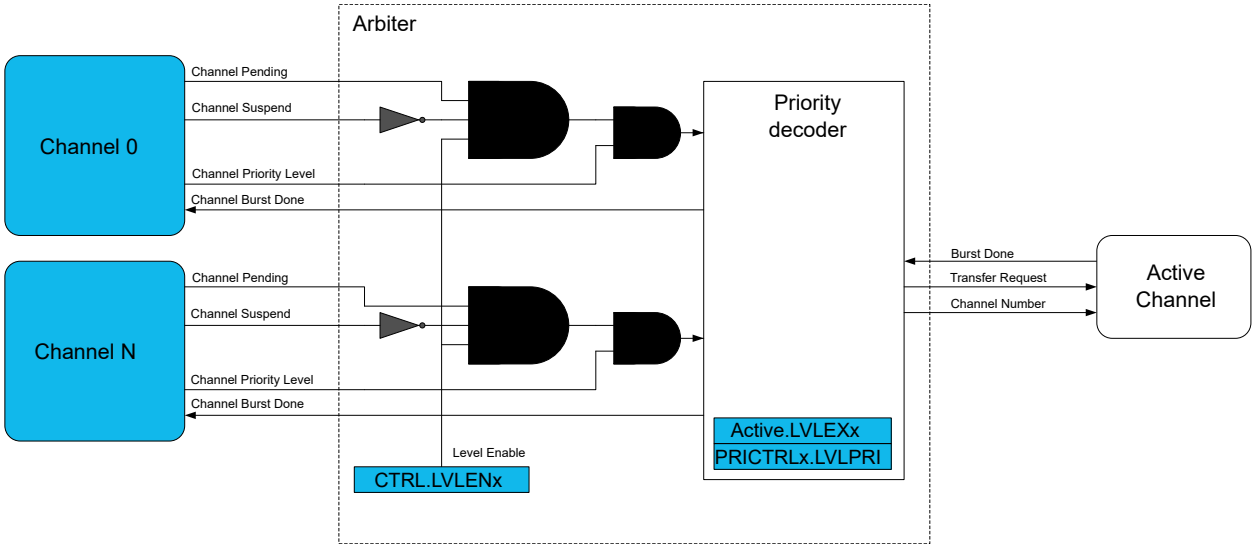
If the upcoming burst transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (13.14.8.12 BUSYCH.BUSYCHx = 1), and it will remain '1' for the subsequent granted burst transfers.

When the channel has performed its granted burst transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended or disabled. This depends on the channel and block transfer configuration. If the DMA channel is fed into the queue of channels with pending transfers, the corresponding 13.14.8.12 BUSYCH.BUSYCHx will remain '1'. If the DMA channel is set to wait for a new transfer trigger, suspended or disabled, the corresponding 13.14.8.12 BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE = 0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 13-50. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

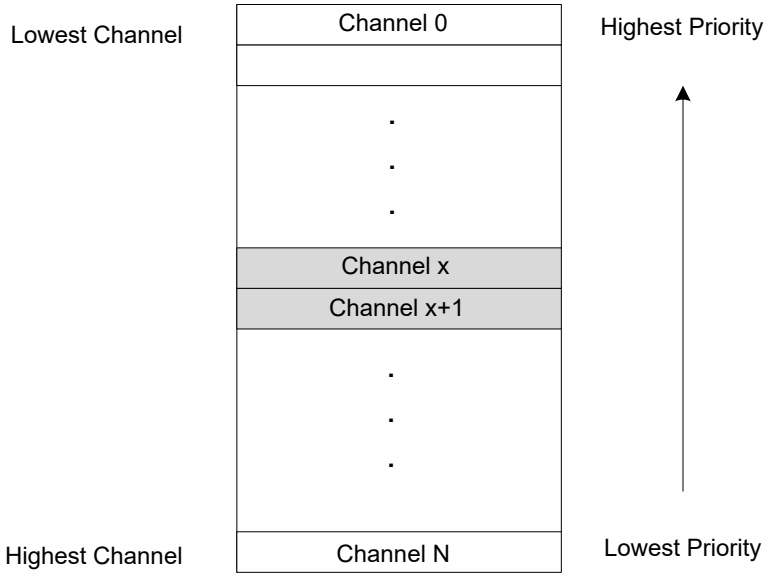
Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (13.14.8.1 CTRL.LVLENx = 1).

Within each priority level, the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control 0 register (PRICTRL0.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration, there is a risk of high channel numbers never being granted access as the active channel. This can be avoided by using a dynamic arbitration scheme.

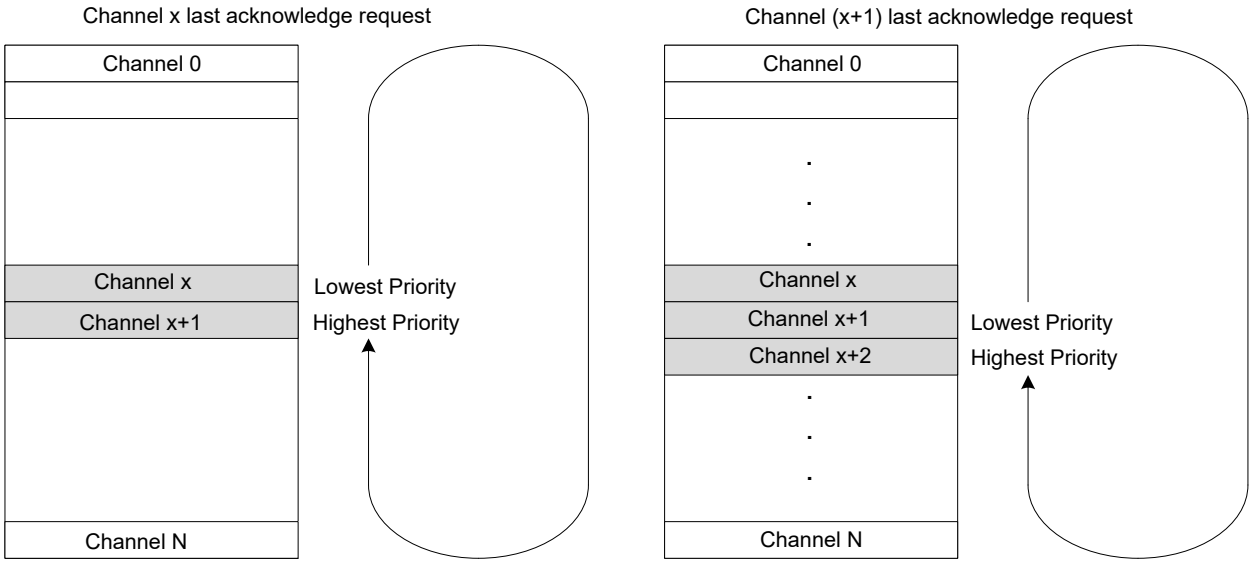
Figure 13-51. Static Priority Scheduling



Dynamic Arbitration within a priority level is selected by writing a '1' to 13.14.8.9 PRICTRL0.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in Figure 13-52. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (13.14.8.9 PRICTRL0.LVLPRIx) for the corresponding priority level.

Figure 13-52. Dynamic (Round-Robin) Priority Scheduling



13.14.6.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to Figure 13-47) the transfer descriptor for the DMA channel will be fetched from LP SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each burst transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a burst, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new burst transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

13.14.6.2.6 Transfer Triggers and Actions

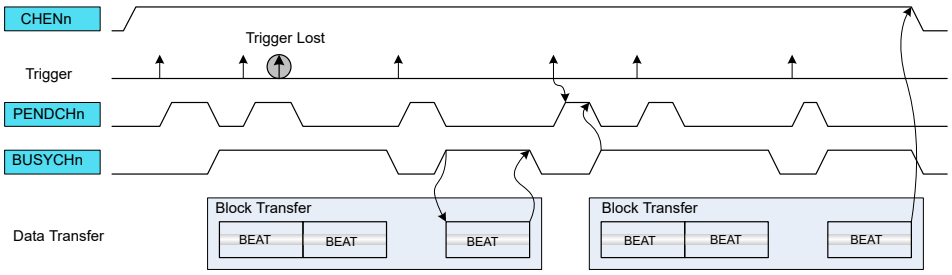
A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected and the DMA channel is granted access to the DMA. A transfer request can be triggered from software, from a peripheral or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer is completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT = 0x2) or transaction transfer (CHCTRLB.TRIGACT = 0x3) instead of a block transfer (CHCTRLB.TRIGACT = 0x0).

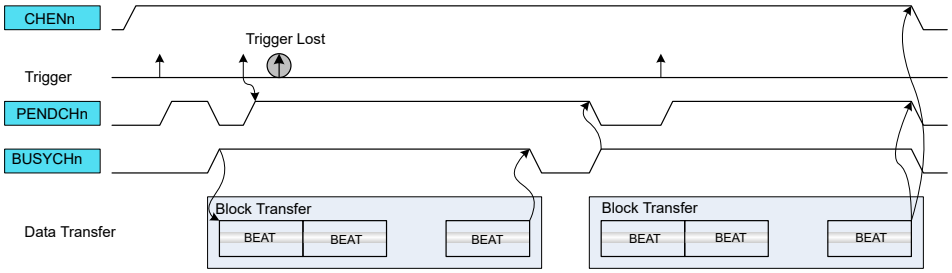
Figure 13-53 shows an example where triggers are used with two linked block descriptors.

Figure 13-53. Trigger Action and Transfers

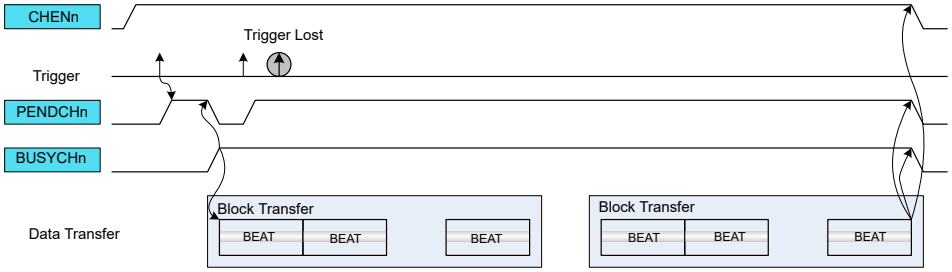
Beat Trigger Action



Block Trigger Action



Transaction Trigger Action



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND = 1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in the Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (13.14.8.12 BUSYCH) in DMAC.

13.14.6.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (SRCADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\text{STEPSIZE}}$$

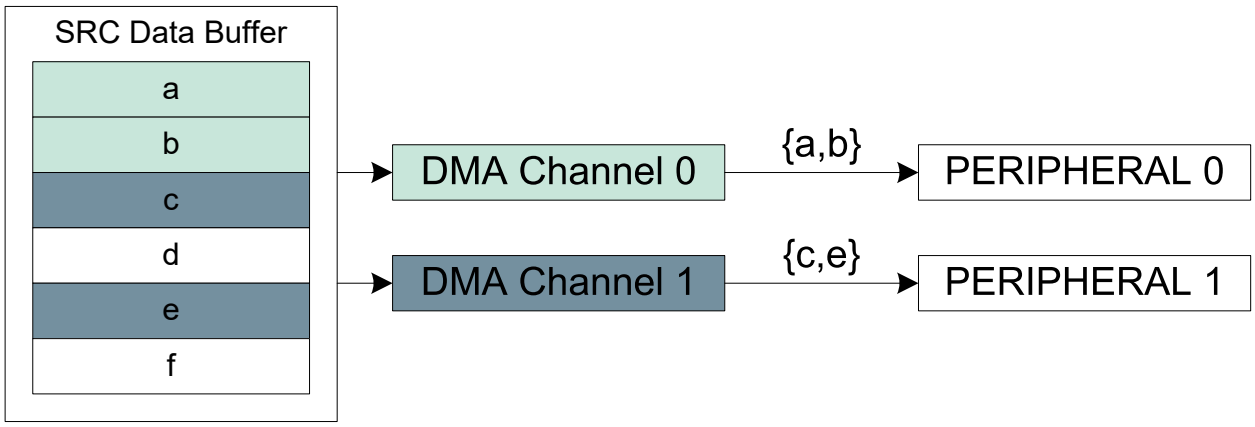
If BTCTRL.STEPSEL=0:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- SRCADDR_{START} is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 13-54. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

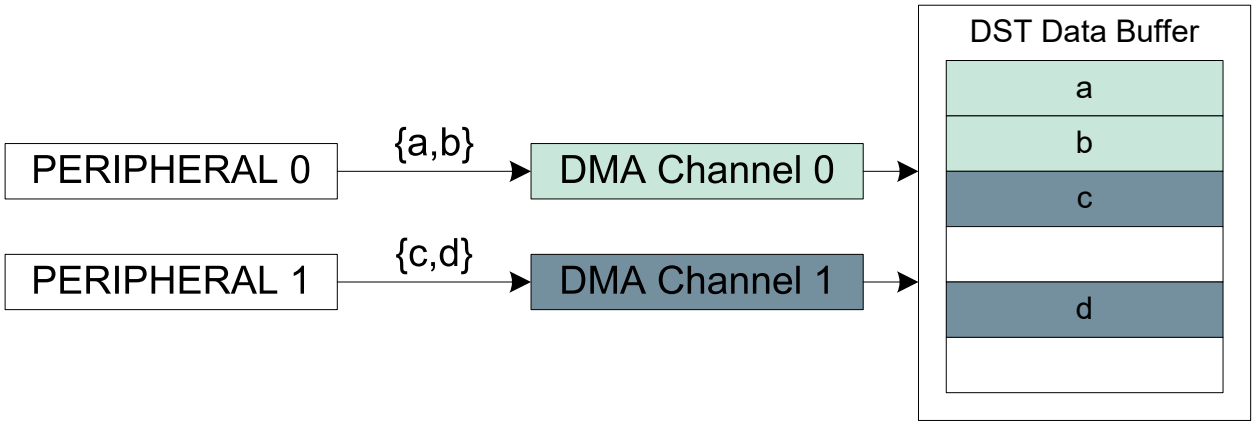
When the destination address incrementation is configured (BTCTRL.DSTINC=1), SRCADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right)$	where BTCTRL.STEPSEL is one

- DSTADDR_{START} is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

Figure 13-55shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0, and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Figure 13-55. Destination Address Increment



13.14.6.2.8 Error Handling

If a bus error is received from an AHB slave during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

13.14.6.3 Additional Features

13.14.6.3.1 Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consist of several block transfers it is called linked descriptors.

Figure Figure 13-49 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from LP SRAM, refer to section 13.14.6.2.5 Data Transmission.

Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in LP SRAM, with DESCADDR=0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in LP SRAM to configure a new descriptor.
4. Configure the new descriptor:
   - Set the next descriptor address (DESCADDR)
   - Set the destination address (DSTADDR)
   - Set the source address (SRCADDR)
   - Configure the block transfer control (BTCTRL) including
     - Optionally enable the Suspend block action
     - Set the descriptor VALID bit

5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the Write-Back memory.
   – If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
      • Update the DESCADDR location of the descriptor from the List
      • Optionally clear the Suspend block action
      • Set the descriptor VALID bit to '1'
      • Optionally enable the Resume software command
   – If the DMA is executing the same descriptor as the one which requires changes:
      • Set the Channel Suspend software command and wait for the Suspend interrupt
      • Update the next descriptor address (DESCRADDR) in the write-back memory
      • Clear the interrupt sources and set the Resume software command
      • Update the DESCADDR location of the descriptor from the List
      • Optionally clear the Suspend block action
      • Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   2.1. Set the descriptor A VALID bit to '0'.
   2.2. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   2.3. Set the DESCADDR value of descriptor C to point to descriptor B.
   2.4. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
   3.1. Apply the software suspend command to the channel and
   3.2. Perform steps 2.1 through 2.4.
   3.3. Apply the software resume command to the channel.

13.14.6.3.2 Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from LP SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

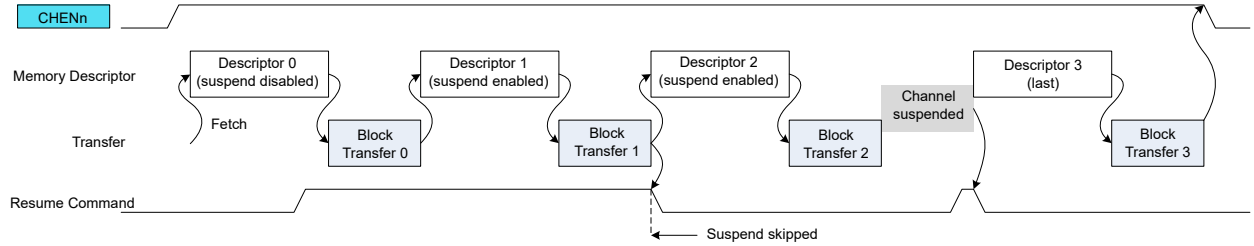
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section 13.14.6.2.3 Transfer Descriptors.

13.14.6.3.3 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 13-56. Channel Suspend/Resume Operation



13.14.6.3.4 Event Input Actions

The event input actions are available only on the least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to 13.14.6.6 Events.

Table 13-40. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	

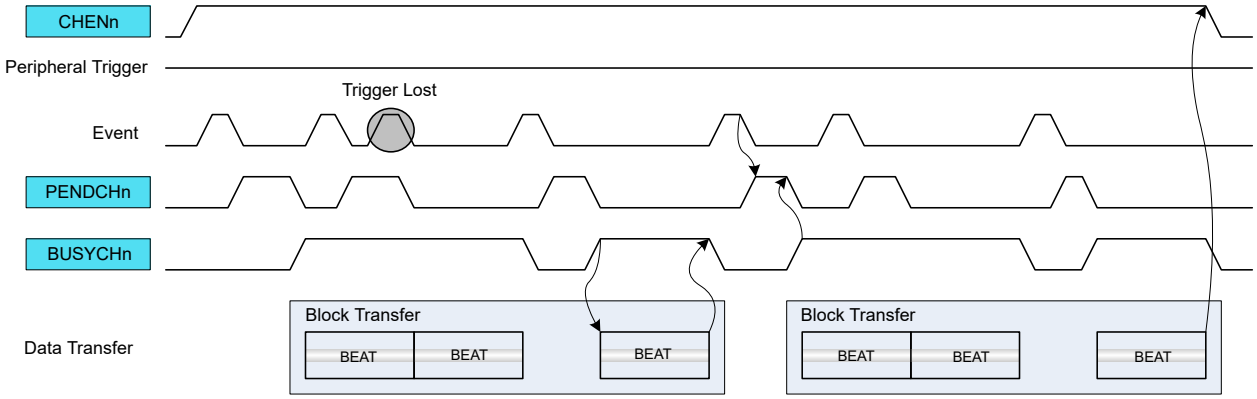
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (13.14.8.13 PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 13-57. Beat Event Trigger Action



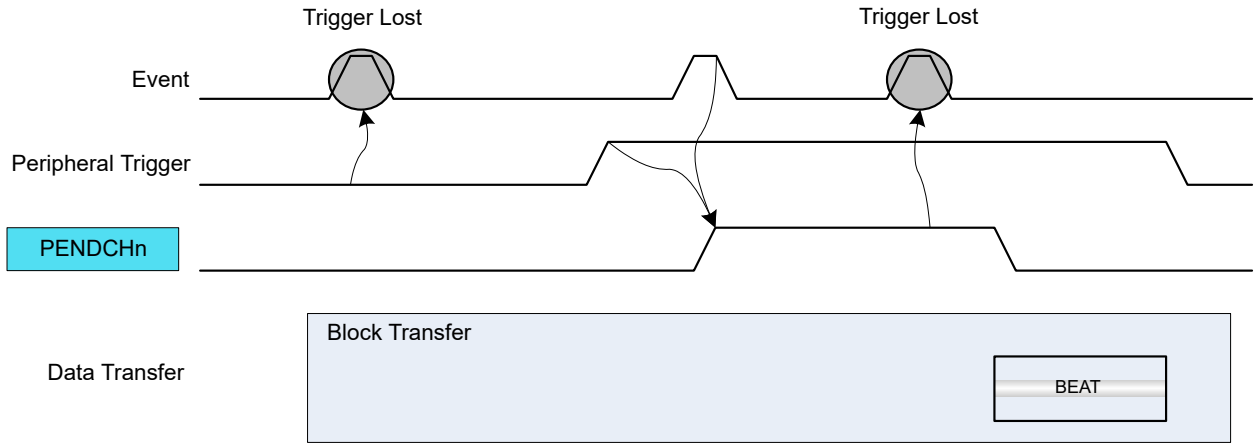
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g. for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and 13.14.8.13 PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 13-58. Periodic Event with Beat Peripheral Triggers



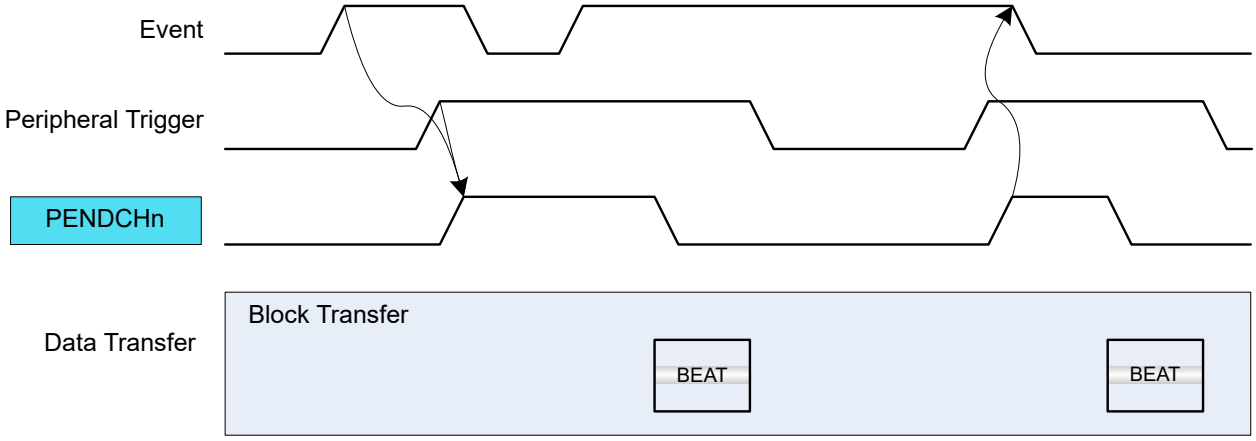
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (13.14.8.13 PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 13-59. Conditional Event with Beat Peripheral Triggers

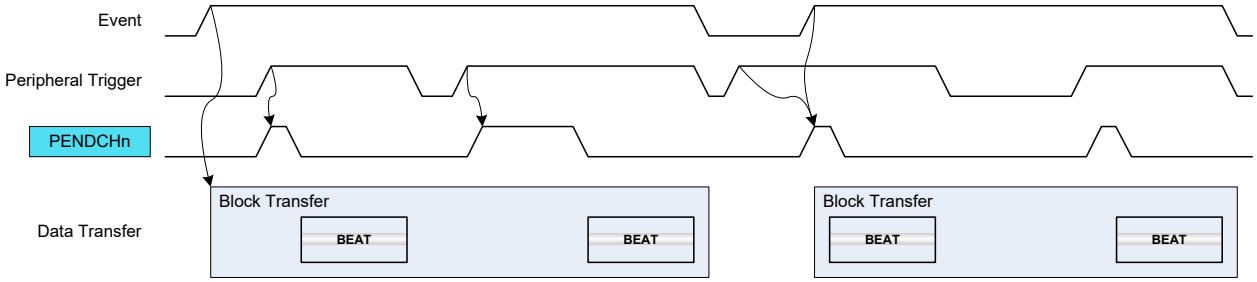


Conditional Block Transfer
The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 13-60. Conditional Block Transfer with Beat Peripheral Triggers



Channel Suspend
The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to 13.14.6.3.2 Channel Suspend.

Channel Resume
The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to 13.14.6.3.2 Channel Suspend.

Skip Next Block Suspend
This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

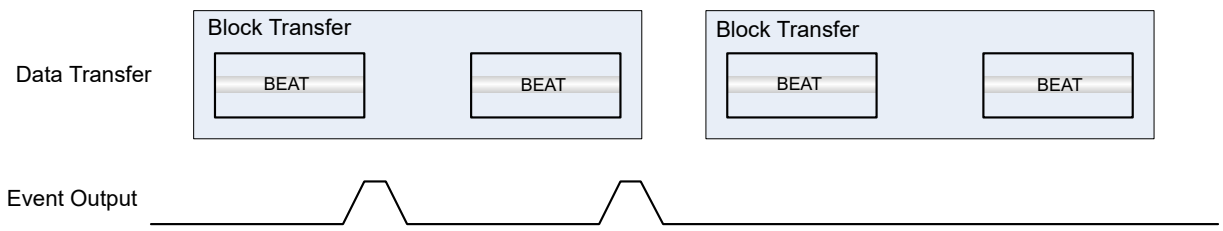
13.14.6.3.5 Event Output Selection
Event output selection is available only for the least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL=0x1) or beat transfer (BTCTRL.EVOSEL=0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

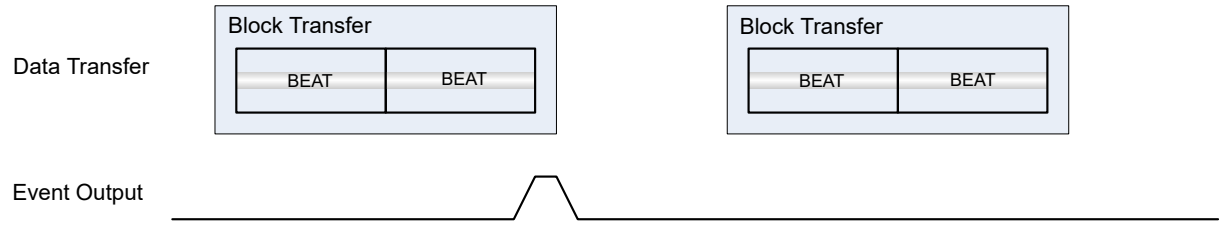
The figure Figure 13-61 shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

Figure 13-61. Event Output Generation

Beat Event Output



Block Event Output



13.14.6.3.6 Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (13.14.8.1 CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

13.14.6.3.7 CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

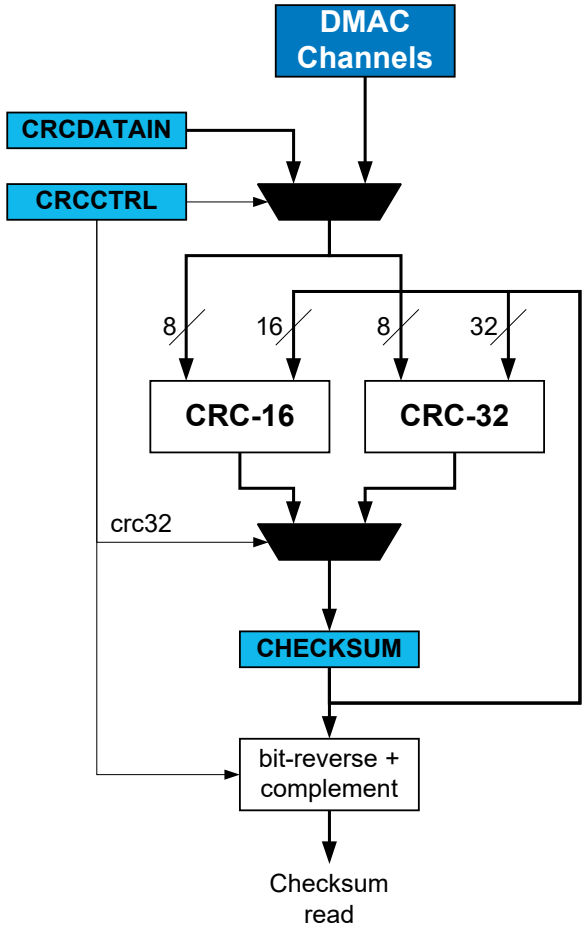
The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 13-62.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 13-62. CRC Generator Block Diagram



CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU,

and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

13.14.6.4 DMA Operation

Not applicable.

13.14.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer completed on the corresponding channel. Refer to 13.14.6.2.5 Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to 13.14.6.2.8 Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel was suspended. Refer to 13.14.6.3.2 Channel Suspend and 13.14.6.2.5 Data Transmission for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET = 1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR = 1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together at the system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (13.14.8.11 INTSTATUS) register to identify the channels with pending interrupts, and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

13.14.6.6 Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to 13.14.6.3.5 Event Output Selection for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition

Setting the Channel Control B Event Input Enable bit (CHCTRLB.EVIE=1) enables the corresponding action on input event. clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to 13.14.6.3.4 Event Input Actions.

Related Links

13.18 EVSYS – Event System

13.14.6.7 Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

For channels with CHCTRLA.RUNSTDBY = 0, it is up to software to stop DMA transfers on these channels and wait for completion before going to standby mode using the following sequence:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes up, resume the suspended channels.

Note: In Stand-by Sleep mode, the DMAC can only access RAM when it is not back biased (PM.STDBYCFG.BBIASxx = 0x0)

Note: In Stand-by Sleep mode, the DMAC can access the LP SRAM only when the power domain PD1 is not in retention and PM.STDBYCFG.BBIASLP=0x0. The DMAC can access the SRAM in Stand-by Sleep mode only when the power domain PD2 is not in retention and PM.STDBYCFG.BBIASHS = 0x0.

13.14.6.8 Synchronization

Not applicable.

13.14.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0							DMAENABLE	SWRST
		15:8					LVLEN3	LVLEN2	LVLEN1	LVLEN0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
		15:8			CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
		15:8	CRCDATAIN[15:8]							
		23:16	CRCDATAIN[23:16]							
		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
		15:8	CRCCHKSUM[15:8]							
		23:16	CRCCHKSUM[23:16]							
		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
		15:8	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
		23:16								
		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0				LVLPRI0[3:0]			
		15:8	RRLVLEN1				LVLPRI1[3:0]			
		23:16	RRLVLEN2				LVLPRI2[3:0]			
		31:24	RRLVLEN3				LVLPRI3[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0					ID[3:0]			
		15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
		15:8	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
		23:16								
		31:24								
0x28	BUSYCH	7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
		15:8	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
		23:16								
		31:24								
0x2C	PENDCH	7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
		15:8	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
		23:16								
		31:24								
0x30	ACTIVE	7:0					LVLEXx	LVLEXx	LVLEXx	LVLEXx
		15:8	ABUSY			ID[4:0]				
		23:16	BTCNT[7:0]							
		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
		15:8	BASEADDR[15:8]							
		23:16	BASEADDR[23:16]							
		31:24	BASEADDR[31:24]							
0x38	WRBADDR	7:0	WRBADDR[7:0]							
		15:8	WRBADDR[15:8]							
		23:16	WRBADDR[23:16]							
		31:24	WRBADDR[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
		15:8			TRIGSRC[5:0]					
		23:16	TRIGACT[1:0]							
		31:24							CMD[1:0]	
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

13.14.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.14.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.14.8.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x00X0
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
					LVLEN3	LVLEN2	LVLEN1	LVLEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							DMAENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bits 8, 9, 10, 11 – LVLENx Priority Level x Enable [x=0..3]
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.
For details on arbitration schemes, refer to the 13.14.6.2.4 Arbitration section.
These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled
1	Transfer requests for Priority level x will be handled

Bit 1 – DMAENABLE DMA Enable
Setting this bit will enable the DMA module.
Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit when both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '0') resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

Value	Description
0	There is no Reset operation ongoing
1	A Reset operation is ongoing

13.14.8.2 CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			CRCSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 13:8 – CRCSRC[5:0] CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	NOACT	No action
0x01	IO	I/O interface
0x02-0x1F	-	Reserved
0x20	CHN	DMA channel 0
0x21	CHN	DMA channel 1
0x22	CHN	DMA channel 2
0x23	CHN	DMA channel 3
0x24	CHN	DMA channel 4
0x25	CHN	DMA channel 5
0x26	CHN	DMA channel 6
0x27	CHN	DMA channel 7
0x28	CHN	DMA channel 8
0x29	CHN	DMA channel 9
0x2A	CHN	DMA channel 10
0x2B	CHN	DMA channel 11
0x2C	CHN	DMA channel 12
0x2D	CHN	DMA channel 13
0x2E	CHN	DMA channel 14
0x2F	CHN	DMA channel 15
0x30	CHN	DMA channel 16
0x31	CHN	DMA channel 17
0x32	CHN	DMA channel 18
0x33	CHN	DMA channel 19
0x34	CHN	DMA channel 20
0x35	CHN	DMA channel 21
0x36	CHN	DMA channel 22
0x37	CHN	DMA channel 23
0x38	CHN	DMA channel 24
0x39	CHN	DMA channel 25
0x3A	CHN	DMA channel 26
0x3B	CHN	DMA channel 27

Value	Name	Description
0x3C	CHN	DMA channel 28
0x3D	CHN	DMA channel 29
0x3E	CHN	DMA channel 30
0x3F	CHN	DMA channel 31

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3		Reserved

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3		Reserved

13.14.8.3 CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0] CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC Checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

13.14.8.4 CRC Checksum

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0] CRC Checksum
These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

13.14.8.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							CRCZERO	CRCBUSY
Access							R	R/W
Reset							0	0

Bit 1 – CRCZERO CRC Zero
This bit is cleared when a new CRC source is selected.
This bit is set when the CRC generation is complete and the CRC Checksum is zero.
When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY CRC Module Busy
This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.
This bit is set when a source configuration is selected and as long as the source is using the CRC module.

13.14.8.6 Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

13.14.8.7 Quality of Service Control

Name: QOSCTRL
Offset: 0x0E
Reset: 0x2A
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	1	0	1	0

Bits 5:4 – DQOS[1:0] Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

DQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0] Fetch Quality of Service
These bits define the memory priority access during the fetch operation.

FQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0] Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

WRBQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Related Links
11.4.3 SRAM Quality of Service

13.14.8.8 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – SWTRIGn Channel n Software Trigger [n = 15..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (13.14.8.23 CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.
This bit is set if 13.14.8.23 CHSTATUS.PEND is already '1' when writing a '1' to that bit.
Writing a '0' to this bit will clear the bit.
Writing a '1' to this bit will generate a DMA software trigger on channel x, if 13.14.8.23 CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.

13.14.8.9 Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3				LVLPRI3[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RRLVLEN2				LVLPRI2[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RRLVLEN1				LVLPRI1[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRLVLEN0				LVLPRI0[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 31 – RRLVLEN3 Level 3 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for the DMA channels with priority level 3. For details on arbitration schemes, refer to 13.14.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 27:24 – LVLPRI3[3:0] Level 3 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN3 = 1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.
When static arbitration is enabled (PRICTRL0.RRLVLEN3 = 0) for priority level 3 and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2 Level 2 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to 13.14.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 19:16 – LVLPRI2[3:0] Level 2 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN2 = 1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.
When static arbitration is enabled (PRICTRL0.RRLVLEN2 = 0) for priority level 2 and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1 Level 1 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to 13.14.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 11:8 – LVLPRI1[3:0] Level 1 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN1 = 1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.
When static arbitration is enabled (PRICTRL0.RRLVLEN1 = 0) for priority level 1 and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0 Level 0 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to 13.14.6.2.4 Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 3:0 – LVLPRI0[3:0] Level 0 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0 = 1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.
When static arbitration is enabled (PRICTRL0.RRLVLEN0 = 0) for priority level 0 and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

13.14.8.10 Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

This register allows the user to identify the lowest DMA channel with pending interrupt.

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – PEND Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Bits 3:0 – ID[3:0] Channel ID
These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.
When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

13.14.8.11 Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CHINTn Channel n Pending Interrupt [n=15..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.
This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

13.14.8.12 Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – BUSYCHn Busy Channel n [x=15..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.
This bit is set when DMA channel n starts a DMA transfer.

13.14.8.13 Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – PENDCH Pending Channel n [n=15..0]
This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on DMA channel n.

13.14.8.14 Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEXx	LVLEXx	LVLEXx	LVLEXx
Access					R	R	R	R
Reset					0	0	0	0

Bits 31:16 – BTCNT[15:0] Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel active busy flag (ABUSY) is set.

Bit 15 – ABUSY Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.
This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0] Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 3,2,1,0 – LVLEXx Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.
This bit is cleared when no request is pending or being executed.

13.14.8.15 Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASEADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASEADDR[31:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

13.14.8.16 Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	WRBADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WRBADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WRBADDR[31:0] Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

13.14.8.17 Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ID[3:0] Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

13.14.8.18 Channel Control A

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY Channel run in standby
This bit is used to keep the DMAC channel running in standby mode.
This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
Writing a '1' to this bit will enable the DMA channel.
This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.14.8.19 Channel Control B

Name: CHCTRLB
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	TRIGACT[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
			TRIGSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0] Software Command
These bits define the software commands. Refer to 13.14.6.3.2 Channel Suspend and 13.14.6.3.3 Channel Resume and Next Suspend Skip.
These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Bits 23:22 – TRIGACT[1:0] Trigger Action
These bits define the trigger action used for a transfer.

TRIGACT[1:0]	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	—	Reserved
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 13:8 – TRIGSRC[5:0] Trigger Source
These bits define the peripheral trigger which is the source of the transfer. For details on trigger selection and trigger modes, refer to 13.14.6.2.6 Transfer Triggers and Actions and CHCTRLB.TRIGACT.

Value	Name	Description
0x00	DISABLE	Only software/event triggers
0x01	SERCOM0 RX	SERCOM0 RX Trigger
0x02	SERCOM0 TX	SERCOM0 TX Trigger
0x03	SERCOM1 RX	SERCOM1 RX Trigger

Value	Name	Description
0x04	SERCOM1 TX	SERCOM1 TX Trigger
0x05	SERCOM2 RX	SERCOM2 RX Trigger
0x06	SERCOM2 TX	SERCOM2 TX Trigger
0x07	SERCOM3 RX	SERCOM3 RX Trigger
0x08	SERCOM3 TX	SERCOM3 TX Trigger
0x09	SERCOM4 RX	SERCOM4 RX Trigger
0x0A	SERCOM4 TX	SERCOM4 TX Trigger
0x0B	TCC0 OVF	TCC0 Overflow Trigger
0x0C	TCC0 MC0	TCC0 Match/Compare 0 Trigger
0x0D	TCC0 MC1	TCC0 Match/Compare 1 Trigger
0x0E	TCC0 MC2	TCC0 Match/Compare 2 Trigger
0x0F	TCC0 MC3	TCC0 Match/Compare 3 Trigger
0x10	TCC1 OVF	TCC1 Overflow Trigger
0x11	TCC1 MC0	TCC1 Match/Compare 0 Trigger
0x12	TCC1 MC1	TCC1 Match/Compare 1 Trigger
0x13	TCC2 OVF	TCC2 Overflow Trigger
0x14	TCC2 MC0	TCC2 Match/Compare 0 Trigger
0x15	TCC2 MC1	TCC2 Match/Compare 1 Trigger
0x16	TC0 OVF	TC0 Overflow Trigger
0x17	TC0 MC0	TC0 Match/Compare 0 Trigger
0x18	TC0 MC1	TC0 Match/Compare 1 Trigger
0x19	TC1 OVF	TC1 Overflow Trigger
0x1A	TC1 MC0	TC1 Match/Compare 0 Trigger
0x1B	TC1 MC1	TC1 Match/Compare 1 Trigger
0x1C	Reserved	Reserved
0x1D	Reserved	Reserved
0x1E	Reserved	Reserved
0x1F	Reserved	Reserved
0x20	Reserved	Reserved
0x21	Reserved	Reserved
0x22	TC4 OVF	TC4 Overflow Trigger
0x23	TC4 MC0	TC4 Match/Compare 0 Trigger
0x24	TC4 MC1	TC4 Match/Compare 1 Trigger
0x25	ADC RESRDY	ADC Result Ready Trigger
0x26	Reserved	Reserved
0x27	Reserved	Reserved
0x28 - 0x2B	Reserved	Reserved
0x2C	Reserved	Reserved
0x2D	Reserved	Reserved

Bits 6:5 – LVL[1:0] Channel Arbitration Level
These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For further details on arbitration schemes, refer to 13.14.6.2.4 Arbitration.
These bits are not enable-protected.

TRIGACT[1:0]	Name	Description
0x0	LVL0	Channel Priority Level 0
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Bit 4 – EVOE Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (13.14.10.1 BTCTRL.EVOSEL).

This bit is available only for the least significant DMA channels. Refer to 13.18.8.8 USERm and 13.18.8.7 CHANNELn of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE Channel Event Input Enable
This bit is available only for the least significant DMA channels. Refer to 13.18.8.8 USERm and 13.18.8.7 CHANNELn of the Event System for details.

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 2:0 – EVACT[2:0] Event Input Action
These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in the CHCTRLB register of the channel is set.
This bit is available only for the least significant DMA channels. Refer to 13.18.8.8 USERm and 13.18.8.7 CHANNELn of the Event System for details.

EVACT[2:0]	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	—	Reserved

13.14.8.20 Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

13.14.8.21 Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

13.14.8.22 Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.
For details on available software commands, refer to CHCTRLB.CMD.
For details on available event input actions, refer to CHCTRLB.EVACT.
For details on available block actions, refer to BTCTRL.BLOCKACT.

Bit 1 – TCMPL Channel Transfer Complete
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR Channel Transfer Error
This flag is cleared by writing a '1' to it.
This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

13.14.8.23 Channel Status

Name: CHSTATUS
Offset: 0x4F
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

Bit 2 – FERR Channel Fetch Error
This bit is cleared when a software resume command is executed.
This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.
This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

13.14.9 Register Summary - LP SRAM

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
		15:8	SRCADDR[15:8]							
		23:16	SRCADDR[23:16]							
		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
		15:8	DSTADDR[15:8]							
		23:16	DSTADDR[23:16]							
		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
		15:8	DESCADDR[15:8]							
		23:16	DESCADDR[23:16]							
		31:24	DESCADDR[31:24]							

13.14.10 Register Description - LP SRAM

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.14.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.14.10.1 Block Transfer Control

Name: BTCTRL
Offset: 0x00
Reset: -
Property: -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset								

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL Step Selection
This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

Bit 10 – SRCINC Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled.
1	The Source Address Increment is enabled.

Bits 9:8 – BEATSIZE[1:0] Beat Size
These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
other		Reserved

Bits 4:3 – BLOCKACT[1:0] Block Action
These bits define what actions the DMAC should take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection
These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID Descriptor Valid
Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.
The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid.
1	The descriptor is valid.

13.14.10.2 Block Transfer Count

Name: BTCNT
Offset: 0x02
Reset: -
Property: -

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	BTCNT[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BTCNT[7:0]							
Access								
Reset								

Bits 15:0 – BTCNT[15:0] Block Transfer Count
This bit group holds the 16-bit block transfer count.
During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

13.14.10.3 Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Reset: -
Property: -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access								
Reset								

Bits 31:0 – SRCADDR[31:0] Transfer Source Address
This bit group holds the source address corresponding to the last beat transfer address in the block transfer.

13.14.10.4 Block Transfer Destination Address

Name: DSTADDR
Offset: 0x08
Reset: -
Property: -

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access								
Reset								

Bits 31:0 – DSTADDR[31:0] Transfer Destination Address
This bit group holds the destination address corresponding to the last beat transfer address in the block transfer.

13.14.10.5 Next Descriptor Address

Name: DESCADDR
Offset: 0x0C
Reset: -
Property: -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access								
Reset								

Bits 31:0 – DESCADDR[31:0] Next Descriptor Address
This bit group holds the LP SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this LP SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

13.15 EIC – External Interrupt Controller

13.15.1 Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, or both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event.

A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

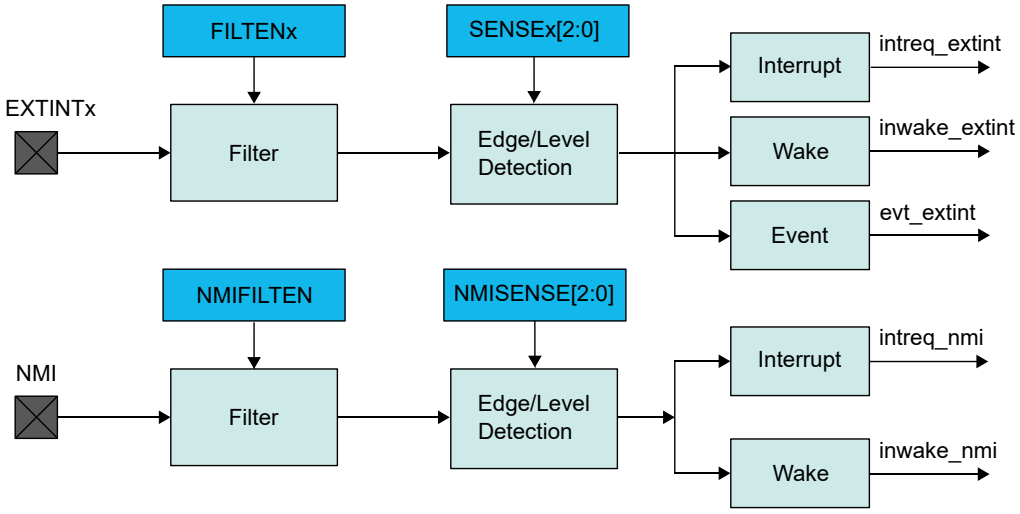
13.15.2 Features

- Up to 16 external pins, plus one non-maskable pin
- Dedicated, individually maskable interrupt for each pin
- Interrupt on rising, falling, or both edges
- Synchronous or asynchronous edge detection mode
- Interrupt on high or low levels
- Asynchronous interrupts for sleep modes without clock
- Filtering of external pins

- Event generation

13.15.3 Block Diagram

Figure 13-63. EIC Block Diagram



13.15.4 Signal Description

Signal Name	Type	Description
EXTINT[15..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal can be mapped on several pins.

13.15.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.15.5.1 I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

Related Links

13.17 PORT - I/O Pin Controller

13.15.5.2 Power Management

All interrupts are available in all sleep modes, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.15.5.3 Clocks

The EIC bus clock (CLK_EIC_APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK_EIC_APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low Power 32KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the Generic Clock Controller.

CLK_ULP32K is provided by the internal ultra-low-power (OSCULP32K) oscillator in the OSC32KCTRL module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (CLK_EIC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

13.6 MCLK – Main Clock

13.6.6.2.6 Peripheral Clock Masking

13.5 GCLK - Generic Clock Controller

13.10 OSC32KCTRL – 32KHz Oscillators Controller

13.15.5.4 DMA

Not applicable.

13.15.5.5 Interrupts

There are two interrupt request lines, one for the external interrupts (EXTINT) and one for non-maskable interrupt (NMI).

The EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is also connected to the interrupt controller, but does not require the interrupt to be configured.

Related Links

11.2 Nested Vector Interrupt Controller

13.15.5.6 Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

13.18 EVSYS – Event System

13.15.5.7 Debug Operation

When the CPU is halted in debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.15.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.15.5.9 Analog Connections

Not applicable.

13.15.6 Functional Description

13.15.6.1 Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

13.15.6.2 Basic Operation

13.15.6.2.1 Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB.
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL).
3. When the NMI is used or synchronous edge detection or filtering are required, enable GCLK_EIC or CLK_ULP32K.
   GCLK_EIC is used when a frequency higher than 32 KHz is required for filtering; CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K, write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL). Optionally, enable the asynchronous mode.
4. Configure the EIC input sense and filtering by writing the Configuration n register (13.15.8.10 CONFIGn).
5. Enable the EIC.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE = 0):

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Configuration n register (13.15.8.10 CONFIGn)
- External Interrupt Asynchronous Mode register (ASYNCH)

Enable-protected bits in the CTRLA register can be written simultaneously while setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

13.15.6.2.2 Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

13.15.6.3 External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIGn.SENSEx) (13.15.8.10 CONFIGn). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag is cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met. In level-sensitive mode, when the interrupt is cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the Configuration n register (CONFIGn.FILTENx) (13.15.8.10 CONFIGn) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

Table 13-41. Majority Vote Filter

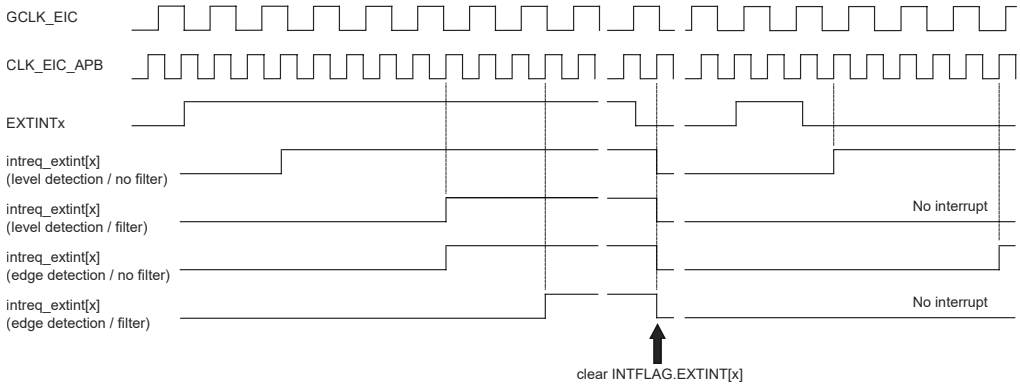
Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0

.....continued	
Samples [0, 1, 2]	**Filter Output**
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Asynchronous detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

If filtering or edge detection is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module.

Figure 13-64. Interrupt Detections



The detection delay depends on the detection mode.

Table 13-42. Interrupt Latency

Detection mode	Latency (worst case)
Level without filter	Five CLK_EIC_APB periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods

Related Links

13.5 GCLK - Generic Clock Controller

13.15.6.4 Additional Features

13.15.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

After reset, NMI is configured to no detection mode.

When an NMI is detected, the non-maskable interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

13.15.6.4.2 Asynchronous Edge Detection Mode

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle sleep mode.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in all sleep modes.

13.15.6.5 DMA Operation

Not applicable.

13.15.6.6 Interrupts

The EIC has the following interrupt sources:

- External interrupt pins (EXTINTx). See 13.15.6.2 Basic Operation.
- Non-maskable interrupt pin (NMI). See 13.15.6.4 Additional Features.

Each interrupt source has an associated interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear interrupt flags. The EIC has one common interrupt request line for all the interrupt sources, and one interrupt request line for the NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Note: If an external interrupts (EXTINT) is common on two or more I/O pins, only one will be active (the first one programmed).

Related Links

11. Processor and Architecture

13.15.6.7 Events

The EIC can generate the following output events:

- External event from pin (EXTINTx).

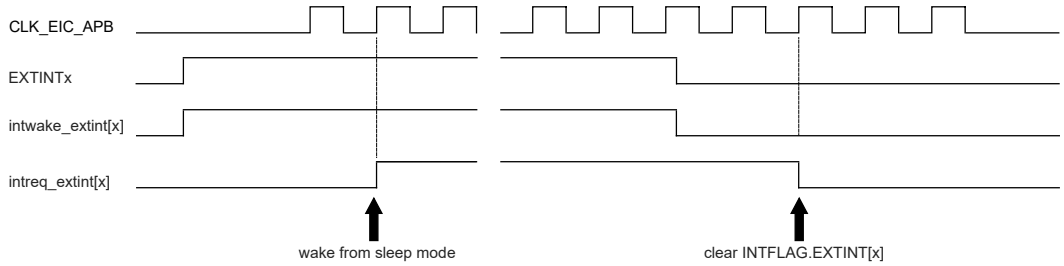
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

13.15.6.8 Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the 13.15.8.10 CONFIGn register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 13-65. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



13.15.6.9 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

13.15.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
		15:8								
0x04	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x08	EVCTRL	7:0	EXTINTEO[7:0]							
		15:8	EXTINTEO[15:8]							
		23:16								
		31:24								
0x0C	INTENCLR	7:0	EXTINT[7:0]							
		15:8	EXTINT[15:8]							
		23:16								
		31:24								
0x10	INTENSET	7:0	EXTINT[7:0]							
		15:8	EXTINT[15:8]							
		23:16								
		31:24								
0x14	INTFLAG	7:0	EXTINT[7:0]							
		15:8	EXTINT[15:8]							
		23:16								
		31:24								
0x18	ASYNCH	7:0	ASYNCH[7:0]							
		15:8	ASYNCH[15:8]							
		23:16								
		31:24								
0x1C	CONFIG0	7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		23:16	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
		31:24	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
0x20	CONFIG1	7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		23:16	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
		31:24	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		

13.15.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.15.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				R/W			R/W	W
Reset				0			0	0

Bit 4 – CKSEL Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).
This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by CLK_ULP32K.

Bit 1 – ENABLE Enable
Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not Enable-Protected.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

13.15.8.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – NMIASYNCH Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0] Non-Maskable Interrupt Sense Configuration
These bits define on which edge or level the NMI triggers.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 – 0x7	-	Reserved

13.15.8.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x02
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NMI
Access								R/W
Reset								0

Bit 0 – NMI Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.
This flag is set when the NMI pin matches the NMI sense configuration, and will generate an interrupt request.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the non-maskable interrupt flag.

13.15.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

13.15.8.5 Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINTEO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINTEO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINTEO[15:0] External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

13.15.8.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

13.15.8.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

13.15.8.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0] External Interrupt
The flag bit x is cleared by writing a '1' to it.
This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the External Interrupt x flag.

13.15.8.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ASYNCH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ASYNCH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ASYNCH[15:0] Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection Mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge detection is synchronously operated.
1	The EXTINT x edge detection is asynchronously operated.

13.15.8.10 External Interrupt Sense Configuration n

Name: CONFIGn
Offset: 0x1C + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 3, 7, 11, 15, 19, 23, 27, 31 – FILTENx Filter Enable x [x=7..0]

Value	Description
0	Filter is disabled for EXTINT[n*8+x] input.
1	Filter is enabled for EXTINT[n*8+x] input.

Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22, 24:26, 28:30 – SENSEx Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for EXTINT[n*8+x] will be generated.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 – 0x7	-	Reserved

13.16 NVMCTRL – Non-Volatile Memory Controller

13.16.1 Overview

Non-Volatile Memory (NVM) is a reprogrammable Flash memory that retains program and data storage even with power off. It embeds a main array and a separate smaller array intended for EEPROM emulation (RWWEE) that can be programmed while reading the main array. The NVM Controller (NVMCTRL) connects to the AHB and APB bus interfaces for system access to the NVM block. The AHB interface is used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

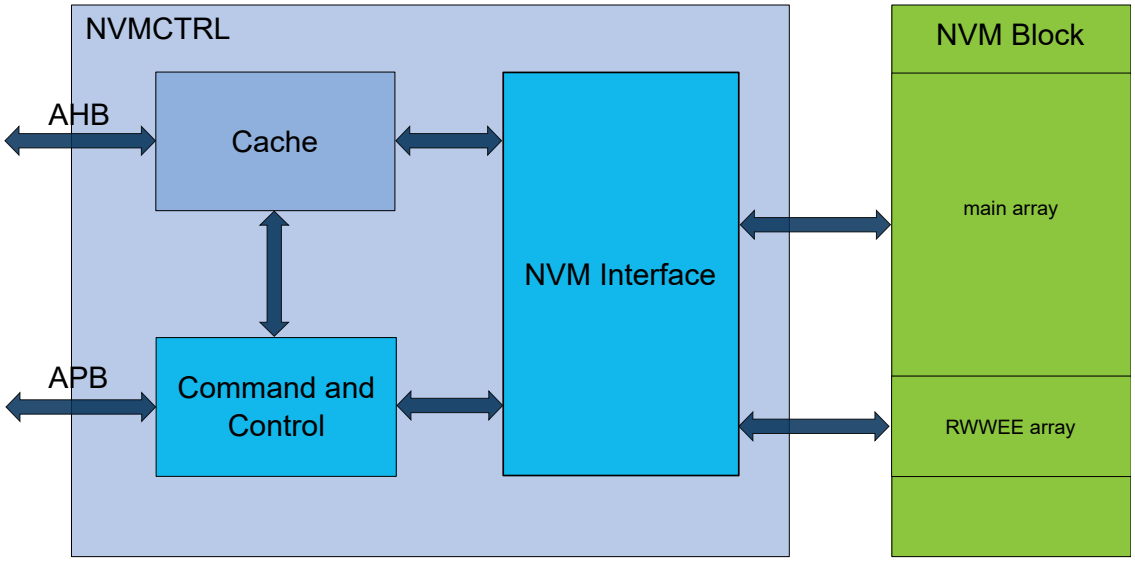
13.16.2 Features

- 32-bit AHB interface for reads and writes
- Read While Write EEPROM emulation area
- All NVM sections are memory mapped to the AHB, including calibration and system configuration
- 32-bit APB interface for commands and control
- Programmable wait states for read optimization
- 16 regions can be individually protected or unprotected
- Additional protection for boot loader
- Supports device protection through a security bit
- Interface to Power Manager for power-down of Flash blocks in sleep modes
- Can optionally wake up on exit from sleep or on first access
- Direct-mapped cache

Note: A register with property "Enable-Protected" may contain bits that are *not* enable-protected.

13.16.3 Block Diagram

Figure 13-66. Block Diagram



13.16.4 Signal Description

Not applicable.

13.16.5 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

13.16.5.1 Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the CTRLB.SLEEPPRM register description for more details.

Related Links

13.8 PM – Power Manager

13.16.5.2 Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range.

Related Links

15. Electrical Characteristics

13.16.5.3 Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

13.16.5.4 Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation.

Access to the NVM block can be protected by the security bit. In this case, the NVM block will not be accessible. See the section on the NVMCTRL 13.16.6.6 Security Bit for details.

13.16.5.5 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC Write-Protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller

13.16.5.6 Analog Connections

Not applicable.

13.16.6 Functional Description

13.16.6.1 Principle of Operation

The NVM Controller is a slave on the AHB and APB buses. It responds to commands, read requests and write requests, based on user configuration.

13.16.6.1.1 Initialization

After power up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

13.16.6.2 Memory Organization

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The NVM is organized into rows, where each row contains four pages, as shown in the NVM Row Organization figure. The NVM has a row-erase granularity, while the write granularity is by page. In other words, a single row erase will erase all four pages in the row, while four write operations are used to write the complete row.

Figure 13-67. NVM Row Organization

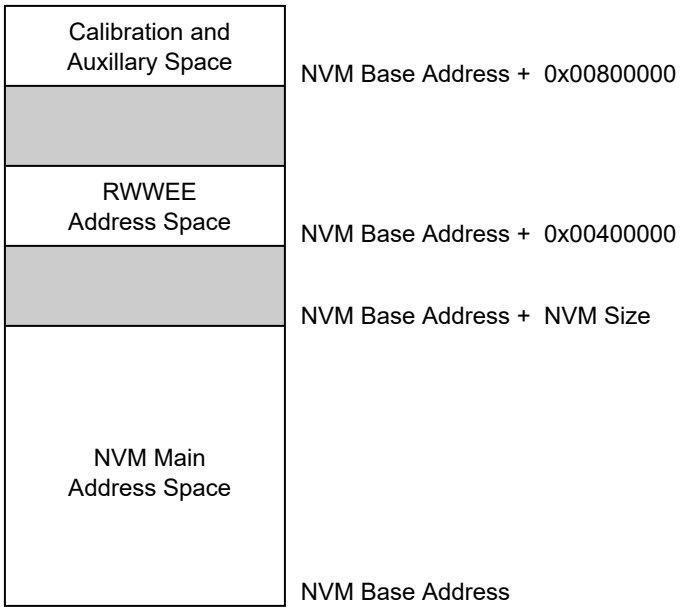
Row n	Page (n*4) + 3	Page (n*4) + 2	Page (n*4) + 1	Page (n*4) + 0

The NVM block contains a calibration and auxiliary space plus a dedicated EEPROM emulation space that are memory mapped. Refer to the NVM Organization figure below for details.

The calibration and auxiliary space contains factory calibration and system configuration information. These spaces can be read from the AHB bus in the same way as the main NVM main address space.

In addition, a boot loader section can be allocated at the beginning of the main array, and an EEPROM section can be allocated at the end of the NVM main address space.

Figure 13-68. NVM Memory Organization

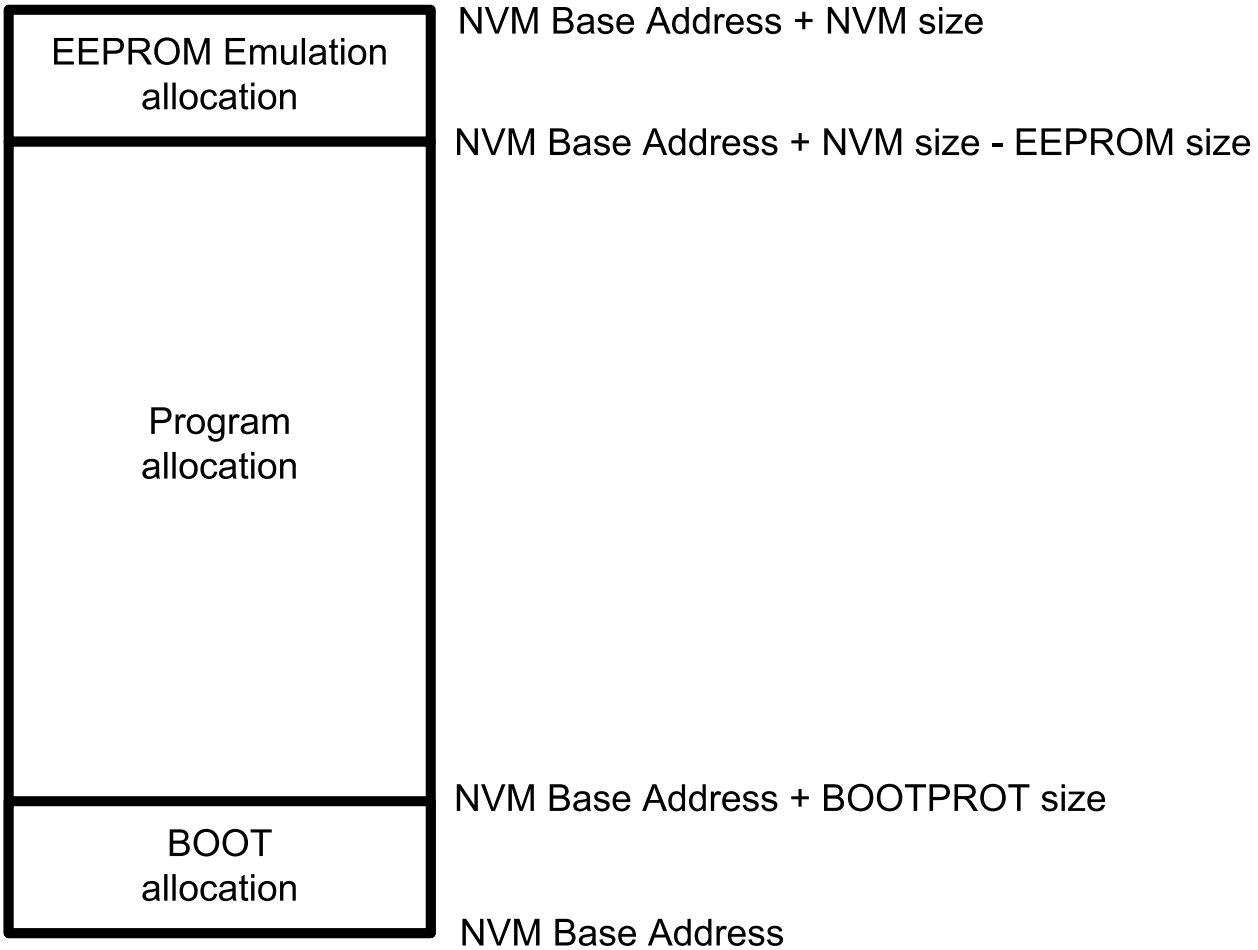


The lower rows in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to EEPROM, as shown in the figure below.

The boot loader section is protected by the lock bit(s) corresponding to this address space and by the BOOTPROT[2:0] fuse. The EEPROM rows can be written regardless of the region lock status.

The number of rows protected by BOOTPROT is given in Boot Loader Size, the number of rows allocated to the EEPROM are given in EEPROM Size.

Figure 13-69. EEPROM and Boot Loader Allocation



Related Links

10.2 Physical Memory Map

13.16.6.3 Region Lock Bits

The NVM block is grouped into 16 equally sized regions. The region size is dependent on the Flash memory size, and is given in the table below. Each region has a dedicated lock bit preventing writing and erasing pages in the region. After production, all regions will be unlocked.

Table 13-43. Region Size

Memory Size [KB]	Region Size [KB]
256	16
128	8
64	4
32	2

To lock or unlock a region, the Lock Region and Unlock Region commands are provided. Writing one of these commands will temporarily lock/unlock the region containing the address loaded in the ADDR register. ADDR can be written by software, or the automatically loaded value from a write operation can be used. The new setting will stay in effect until the next Reset, or until the setting is changed again using the Lock and Unlock commands. The current status of the lock can be determined by reading the LOCK register.

To change the default lock/unlock setting for a region, the user configuration section of the auxiliary space must be written using the Write Auxiliary Page command. Writing to the auxiliary space will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

13.16.6.4 Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the NVM main address space or the RWWEE address space directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, INTFLAG.READY will be cleared until the command has completed. Any commands written while INTFLAG.READY is low will be ignored.

Read the CTRLA register description for more details.

The CTRLB register must be used to control the power reduction mode, read wait states, and the write mode.

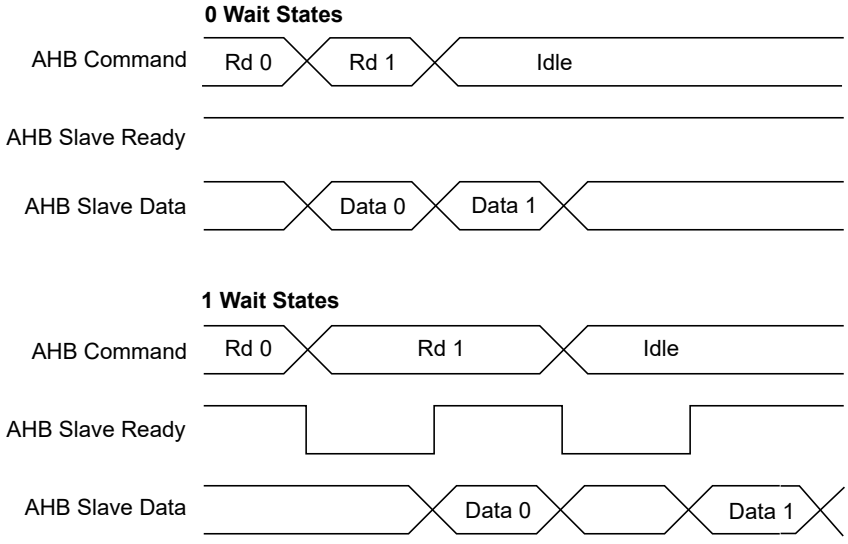
13.16.6.4.1 NVM Read

Reading from the NVM main address space is performed via the AHB bus by addressing the NVM main address space or auxiliary address space directly. Read data is available after the configured number of read wait states (CTRLB.RWS) set in the NVM Controller.

The number of cycles data are delayed to the AHB bus is determined by the read wait states. Examples of using zero and one wait states are shown in Figure Read Wait State Examples below.

Reading the NVM main address space while a programming or erase operation is ongoing on the NVM main array results in an AHB bus stall until the end of the operation. Reading the NVM main array does not stall the bus when the RWWEE array is being programmed or erased.

Figure 13-70. Read Wait State Examples



13.16.6.4.2 RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly. Refer to the figures in 13.16.6.2 Memory Organization for details.

Read timings are similar to regular NVM read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the RWWEE area while the NVM main array is being written or erased, whereas the RWWEE area can be written or erased while the main array is being read.

The RWWEE address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

13.16.6.4.3 NVM Write

The NVM Controller requires that an erase must be done before programming. The entire NVM main address space and the RWWEE address space can be erased by a debugger Chip Erase command. Alternatively, rows can be individually erased by the Erase Row command or the RWWEE Erase Row command to erase the NVM main address space or the RWWEE address space, respectively.

After programming the NVM main array, the region that the page resides in can be locked to prevent spurious write or erase sequences. Locking is performed on a per-region basis, and so, locking a region will lock all pages inside the region.

Data to be written to the NVM block are first written to and stored in an internal buffer called the *page buffer*. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a system exception.

Both the NVM main array and the RWWEE array share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the NVM main array or the RWWEE array by setting CTRLA.CMD to 'Write Page' or 'RWWEE Write Page', respectively, and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not. Before writing the page to memory, the accessed row must be erased.

Automatic page writes are enabled by writing the manual write bit to zero (CTRLB.MANW=0). This will trigger a write operation to the page addressed by ADDR when the last location of the page is written.

Because the address is automatically stored in ADDR during the I/O bus write operation, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written.

Procedure for Manual Page Writes (CTRLB.MANW=1)

The row to be written to must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly
- Write the page buffer to memory: CTRL.CMD='Write Page' and CMDEX
- The READY bit in the INTFLAG register will be low while programming is in progress, and access through the AHB will be stalled

Procedure for Automatic Page Writes (CTRLB.MANW=0)

The row to be written to must be erased before the last write to the page buffer is performed.

Note that partially written pages must be written with a manual write.

- Write to the page buffer by addressing the NVM main address space directly.
  When the last location in the page buffer is written, the page is automatically written to NVM main address space.
- INTFLAG.READY will be zero while programming is in progress and access through the AHB will be stalled.

13.16.6.4.4 Page Buffer Clear

The page buffer is automatically set to all '1' after a page write is performed. If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear command can be used.

13.16.6.4.5 Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM main address space. The RWWEE Erase Row can be used to erase the desired row in the RWWEE array. Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the Status register (STATUS.LOCKE) will be set.

Procedure for Erase Row

- Write the address of the row to erase to ADDR. Any address within the row can be used.
- Issue an Erase Row command.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

13.16.6.4.6 Lock and Unlock Region

These commands are used to lock and unlock regions as detailed in section 13.16.6.3 Region Lock Bits.

13.16.6.4.7 Set and Clear Power Reduction Mode

The NVM Controller and block can be taken in and out of power reduction mode through the Set and Clear Power Reduction Mode commands. When the NVM Controller and block are in power reduction mode, the Power Reduction Mode bit in the Status register (STATUS.PRM) is set.

13.16.6.5 NVM User Configuration

The NVM user configuration resides in the auxiliary space. Refer to the Physical Memory Map of the device for calibration and auxiliary space address mapping.

The bootloader resides in the main array starting at offset zero. The allocated boot loader section is write-protected.

Table 13-44. Boot Loader Size

BOOTPROT [2:0]	Rows Protected by BOOTPROT	Boot Loader Size in Bytes
0x7(1)	None	0
0x6	2	512
0x5	4	1024
0x4	8	2048
0x3	16	4096
0x2	32	8192
0x1	64	16384
0x0	128	32768

Note: 1) Default value is 0x7.

The EEPROM[2:0] bits indicate the EEPROM size, see the table below. The EEPROM resides in the upper rows of the NVM main address space and is writable, regardless of the region lock status.

Table 13-45. EEPROM Size

EEPROM[2:0]	Rows Allocated to EEPROM	EEPROM Size in Bytes
7	None	0
6	1	256
5	2	512
4	4	1024
3	8	2048
2	16	4096
1	32	8192
0	64	16384

Related Links

10.2 Physical Memory Map

13.16.6.6 Security Bit

The security bit allows the entire chip to be locked from external access for code security. The security bit can be written by a dedicated command, Set Security Bit (SSB). Once set, the only way to clear the security bit is through a debugger Chip Erase command. After issuing the SSB command, the PROGE error bit can be checked.

In order to increase the security level it is recommended to enable the internal BOD33 when the security bit is set.

Related Links

13.3 DSU - Device Service Unit

13.16.6.7 Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the NVM main array address space is cached. It is a direct-mapped cache that implements 64 lines of 64 bits (i.e., 512 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register (CTRLB.CACHEDIS).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register (CTRLB.READMODE).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines (CTRLA.CMD=INVALL). Commands affecting NVM content automatically invalidate cache lines.

13.16.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		CMD[6:0]						
		15:8	CMDEX[7:0]							
0x02 ... 0x03	Reserved									
0x04	CTRLB	7:0	MANW			RWS[3:0]				
		15:8							SLEEPPRM[1:0]	
		23:16						CACHEDIS	READMODE[1:0]	
		31:24								
0x08	PARAM	7:0	NVMP[7:0]							
		15:8	NVMP[15:8]							
		23:16	RWWEEP[3:0]					PSZ[2:0]		
		31:24	RWWEEP[11:4]							
0x0C	INTENCLR	7:0							ERROR	
0x0D ... 0x0F	Reserved									
0x10	INTENSET	7:0							ERROR	
0x11 ... 0x13	Reserved									
0x14	INTFLAG	7:0							ERROR	
0x15 ... 0x17	Reserved									
0x18	STATUS	7:0				NVME	LOCKE	PROGE	LOAD	PRM
		15:8								SB
0x1A ... 0x1B	Reserved									
0x1C	ADDR	7:0	ADDR[7:0]							
		15:8	ADDR[15:8]							
		23:16			ADDR[21:16]					
		31:24								
0x20	LOCK	7:0	LOCK[7:0]							
		15:8	LOCK[15:8]							

13.16.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.16.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CMD[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 15:8 – CMDEX[7:0] Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the Programming Error bit in the Status register (STATUS.PROGE) will be set. PROGE is also set if a previously written command is not completed yet.
The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.
INTFLAG.READY must be '1' when the command is issued.
Bit 0 of the CMDEX bit group will read back as '1' until the command is issued.
Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

Bits 6:0 – CMD[6:0] Command
These bits define the command to be executed when the CMDEX key is written.

CMD[6:0]	Group Configuration	Description
0x00-0x01	-	Reserved
0x02	ER	Erase Row - Erases the row addressed by the ADDR register in the NVM main array.
0x03	-	Reserved
0x04	WP	Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register.
0x05	EAR	Erase Auxiliary Row - Erases the auxiliary row addressed by the ADDR register. This command can be given only when the security bit is not set and only to the User Configuration Row.
0x06	WAP	Write Auxiliary Page - Writes the contents of the page buffer to the page addressed by the ADDR register. This command can be given only when the security bit is not set and only to the User Configuration Row.
0x07-0x0E	-	Reserved
0x0F	WL	Write Lockbits- write the LOCK register
0x1A-0x19	-	Reserved
0x1A	RWWEEER	RWWEE Erase Row - Erases the row addressed by the ADDR register in the RWWEE array.
0x1B	-	Reserved
0x1C	RWWEEWP	RWWEE Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register in the RWWEE array.
0x1D-0x3F	-	Reserved
0x40	LR	Lock Region - Locks the region containing the address location in the ADDR register.

.....continued		
CMD[6:0]	**Group Configuration**	**Description**
0x41	UR	Unlock Region - Unlocks the region containing the address location in the ADDR register.
0x42	SPRM	Sets the Power Reduction Mode.
0x43	CPRM	Clears the Power Reduction Mode.
0x44	PBC	Page Buffer Clear - Clears the page buffer.
0x45	SSB	Set Security Bit - Sets the security bit by writing 0x00 to the first byte in the lockbit row.
0x46	INVALL	Invalidates all cache lines.
0x47	LDR	Lock Data Region - Locks the data region containing the address location in the ADDR register. When the Security Extension is enabled, only secure access can lock secure regions.
0x48	UDR	Unlock Data Region - Unlocks the data region containing the address location in the ADDR register. When the Security Extension is enabled, only secure access can unlock secure regions.
0x47-0x7F	-	Reserved

13.16.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CACHEDIS	READMODE[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							SLEEPPRM[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	MANW			RWS[3:0]				
Access	R/W			R/W	R/W	R/W	R/W	
Reset	1			0	0	0	0	

Bit 18 – CACHEDIS Cache Disable
This bit is used to disable the cache.

Value	Description
0	The cache is enabled
1	The cache is disabled

Bits 17:16 – READMODE[1:0] NVMCTRL Read Mode

Value	Name	Description
0x0	NO_MISS_PENALTY	The NVM Controller (cache system) does not insert wait states on a cache miss. Gives the best system performance.
0x1	LOW_POWER	Reduces power consumption of the cache system, but inserts a wait state each time there is a cache miss. This mode may not be relevant if CPU performance is required, as the application will be stalled and may lead to increased run time.
0x2	DETERMINISTIC	The cache system ensures that a cache hit or miss takes the same amount of time, determined by the number of programmed Flash wait states. This mode can be used for real-time applications that require deterministic execution timings.
0x3	Reserved	

Bits 9:8 – SLEEPPRM[1:0] Power Reduction Mode during Sleep
Indicates the Power Reduction Mode during sleep.

Value	Name	Description
0x0	WAKEUPACCESS	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode upon first access.
0x1	WAKEUPINSTANT	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode when exiting sleep.
0x2	Reserved	
0x3	DISABLED	Auto power reduction disabled.

Bit 7 – MANW Manual Write
Note that reset value of this bit is '1'.

Value	Description
0	Writing to the last word in the page buffer will initiate a write operation to the page addressed by the last write operation. This includes writes to memory and auxiliary rows.
1	Write commands must be issued through the CTRLA.CMD register.

Bits 4:1 – RWS[3:0] NVM Read Wait States
These bits control the number of wait states for a read operation. '0' indicates zero wait states, '1' indicates one wait state, etc., up to 15 wait states.
This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

13.16.8.3 NVM Parameter

Name: PARAM
Offset: 0x08
Reset: 0x000XXXXX
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RWWEEP[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RWWEEP[3:0]					PSZ[2:0]		
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		x	x	x

Bit	15	14	13	12	11	10	9	8
	NVMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	NVMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:20 – RWWEEP[11:0] Read While Write EEPROM emulation area Pages
Indicates the number of pages in the RWW EEPROM emulation address space.

Bits 18:16 – PSZ[2:0] Page Size
Indicates the page size. Not all devices of the device families will provide all the page sizes indicated in the table.

Value	Name	Description
0x0	8	8 bytes
0x1	16	16 bytes
0x2	32	32 bytes
0x3	64	64 bytes
0x4	128	128 bytes
0x5	256	256 bytes
0x6	512	512 bytes
0x7	1024	1024 bytes

Bits 15:0 – NVMP[15:0] NVM Pages
Indicates the number of pages in the NVM main address space.

13.16.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
							ERROR	
Access							R/W	
Reset							0	

Bit 1 – ERROR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the ERROR interrupt enable.
This bit will read as the current value of the ERROR interrupt enable.

13.16.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
							ERROR	
Access							R/W	
Reset							0	

Bit 1 – ERROR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the ERROR interrupt enable.
This bit will read as the current value of the ERROR interrupt enable.

13.16.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							ERROR	
Access							R/W	
Reset							0	

Bit 1 – ERROR Error
This flag is set on the occurrence of an NVME, LOCKE or PROGE error.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No errors have been received since the last clear.
1	At least one error has occurred since the last clear.

13.16.8.7 Status

Name: STATUS
Offset: 0x18
Reset: 0x0X00
Property: –

Bit	15	14	13	12	11	10	9	8
								SB
Access								R
Reset								x

Bit	7	6	5	4	3	2	1	0
				NVME	LOCKE	PROGE	LOAD	PRM
Access				R/W	R/W	R/W	R/W	R
Reset				0	0	0	0	0

Bit 8 – SB Security Bit Status

Value	Description
0	The Security bit is inactive.
1	The Security bit is active.

Bit 4 – NVME NVM Error
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming or erase errors have been received from the NVM controller since this bit was last cleared.
1	At least one error has been registered from the NVM Controller since this bit was last cleared.

Bit 3 – LOCKE Lock Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming of any locked lock region has happened since this bit was last cleared.
1	Programming of at least one locked lock region has happened since this bit was last cleared.

Bit 2 – PROGE Programming Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No invalid commands or bad keywords were written in the NVM Command register since this bit was last cleared.
1	An invalid command and/or a bad keyword was/were written in the NVM Command register since this bit was last cleared.

Bit 1 – LOAD NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set. It remains set until a page write or a page buffer clear (PBCLR) command is given.
This bit can be cleared by writing a '1' to its bit location.

Bit 0 – PRM Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with SLEEPPRM set accordingly.
PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.

Value	Description
1	NVM is in power reduction mode.

13.16.8.8 Address

Name: ADDR
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			ADDR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 21:0 – ADDR[21:0] NVM Address
ADDR drives the hardware (16-bit) address to the NVM when a command is executed using CMDEX. This register is also automatically updated when writing to the page buffer.

13.16.8.9 Lock Section

Name: LOCK
Offset: 0x20
Reset: 0xXXXX
Property: –

Bit	15	14	13	12	11	10	9	8
	LOCK[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOCK[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	x

Bits 15:0 – LOCK[15:0] Region Lock Bits
In order to set or clear these bits, the CMD register must be used.
Default state after erase will be unlocked (0x0000).

Value	Description
0	The corresponding lock region is locked.
1	The corresponding lock region is not locked.

13.17 PORT - I/O Pin Controller

13.17.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

The PORT is connected to the high-speed bus matrix through an AHB/APB bridge. The Pin Direction, Data Output Value and Data Input Value registers may also be accessed using the low-latency CPU local bus (IOBUS; ARM® single-cycle I/O port).

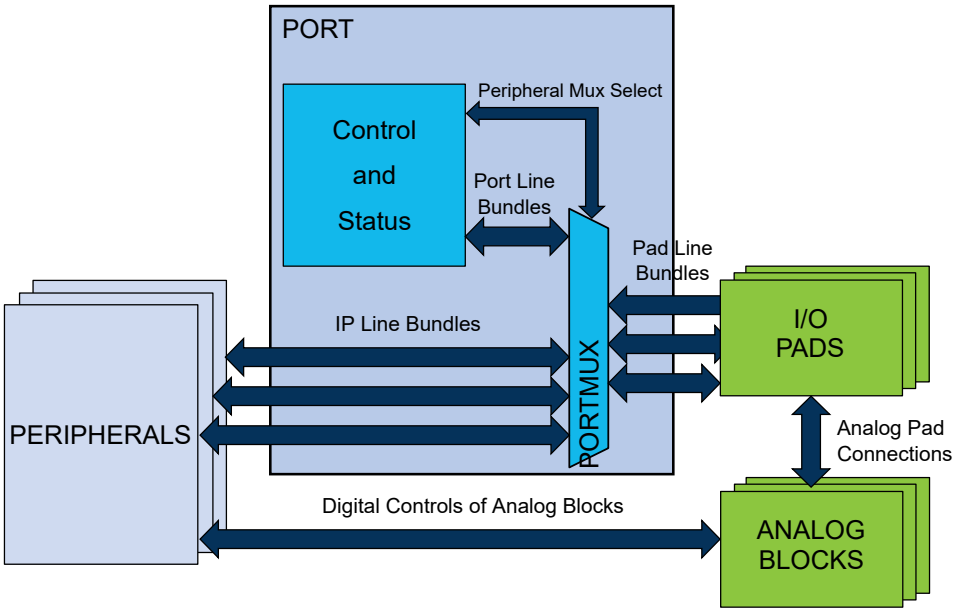
13.17.2 Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria

  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin
- Power saving using STANDBY mode
  - No access to configuration registers
  - Possible access to data registers (DIR, OUT or IN)

13.17.3 Block Diagram

Figure 13-71. PORT Block Diagram



13.17.4 Signal Description

Table 13-46. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.17.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly as following.

13.17.5.1 I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral has control over

the output state of the pad, as well as the ability to read the current physical pad state. Refer to *I/O Multiplexing and Considerations* for details.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

Related Links

7. I/O Multiplexing and Considerations

13.17.5.2 Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

When the device is set to the BACKUP sleep mode, even if the PORT configuration registers and input synchronizers will lose their contents (these will not be restored when PORT is powered up again), the latches in the pads will keep their current configuration, such as the output value and pull settings. Refer to the Power Manager documentation for more features related to the I/O lines configuration in and out of BACKUP mode.

The PORT peripheral will continue operating in any sleep mode where its source clock is running.

13.17.5.3 Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT is fed by two different clocks: a CPU main clock, which allows the CPU to access the PORT through the low latency CPU local bus (IOBUS); an APB clock, which is a divided clock of the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The priority of IOBUS accesses is higher than event accesses and APB accesses. The EVSYS and APB will insert wait states in the event of concurrent PORT accesses.

The PORT input synchronizers use the CPU main clock so that the resynchronization delay is minimized with respect to the APB clock.

Related Links

13.6 MCLK – Main Clock

13.6.6.2.6 Peripheral Clock Masking

13.17.5.4 DMA

Not applicable.

13.17.5.5 Interrupts

Not applicable.

13.17.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

13.18 EVSYS – Event System

13.17.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.17.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.17.5.9 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

13.17.5.10 CPU Local Bus

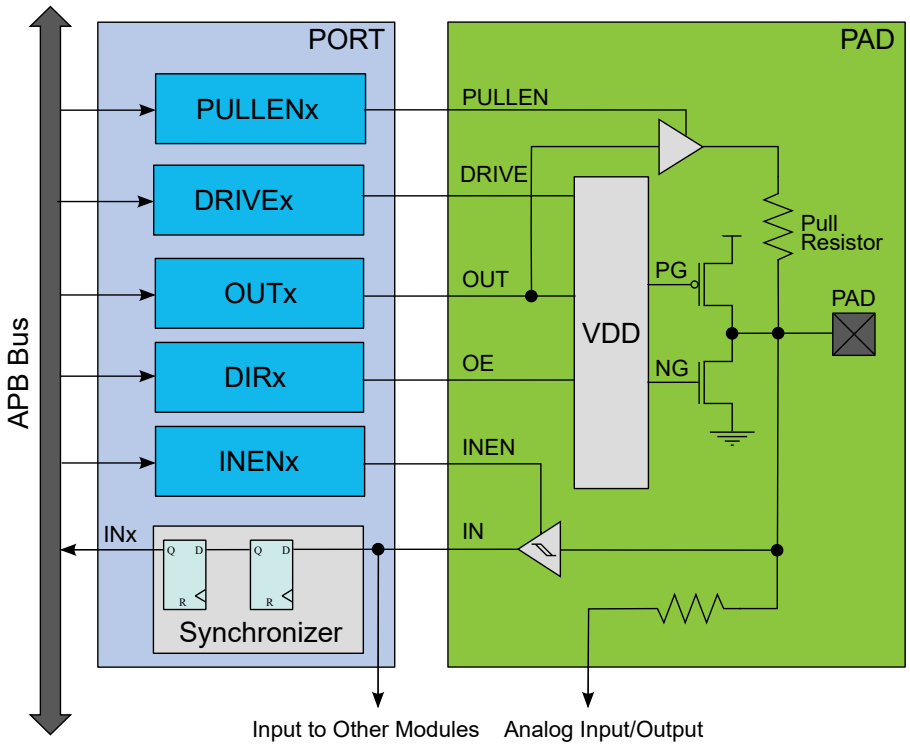
The CPU local bus (IOBUS) is an interface that connects the CPU directly to the PORT. It is a single-cycle bus interface, which does not support wait states. It supports 8-bit, 16-bit and 32-bit sizes.

This bus is generally used for low latency operation. The Data Direction (DIR) and Data Output Value (OUT) registers can be read, written, set, cleared or be toggled using this bus, and the Data Input Value (IN) registers can be read.

Since the IOBUS cannot wait for IN register resynchronization, the Control register (CTRL) must be configured to continuous sampling of all pins that need to be read via the IOBUS in order to prevent stale data from being read.

13.17.6 Functional Description

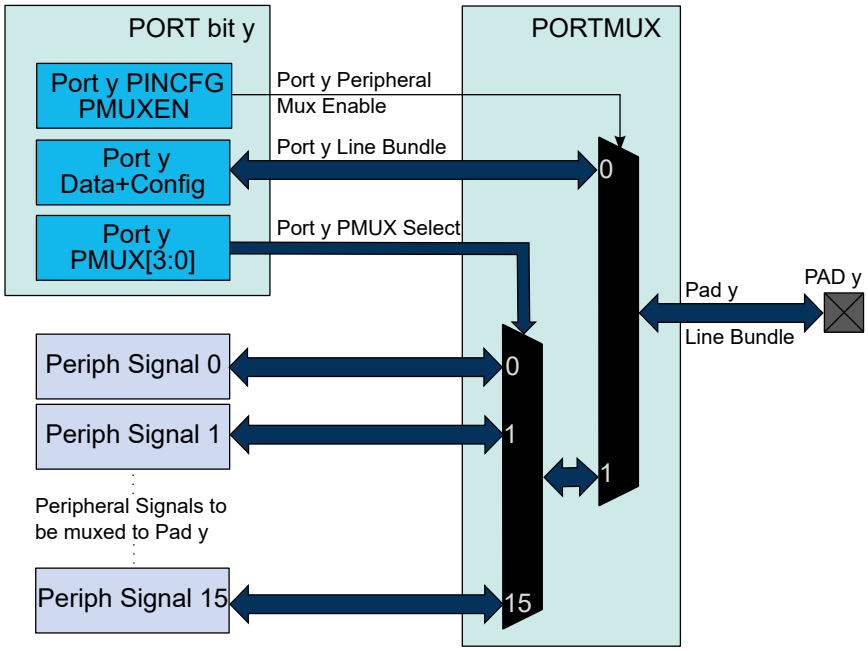
Figure 13-72. Overview of the PORT



13.17.6.1 Principle of Operation

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses. The number of PORT groups may depend on the package/number of pins.

Figure 13-73. Overview of the peripheral functions multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

13.17.6.2 Basic Operation

13.17.6.2.1 Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

13.17.6.2.2 Operation

Each I/O pin y can be controlled by the registers in PORT. Each PORT group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'.

Refer to *I/O Multiplexing and Considerations* for details on pin configuration and PORT groups.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

To use pin y as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin y is at byte offset (PINCFG0 + [y]).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

Related Links

7. I/O Multiplexing and Considerations

13.17.6.3 I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 13-47.

13.17.6.3.1 Pin Configurations Summary

Table 13-47. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

13.17.6.3.2 Input Configuration

Figure 13-74. I/O configuration - Standard Input

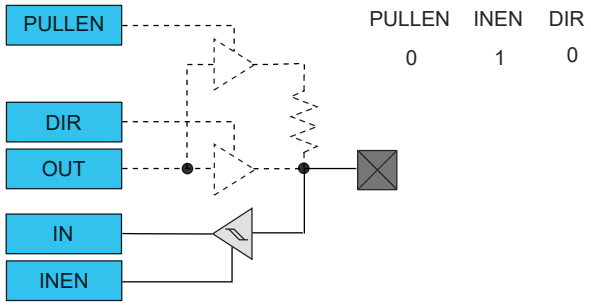
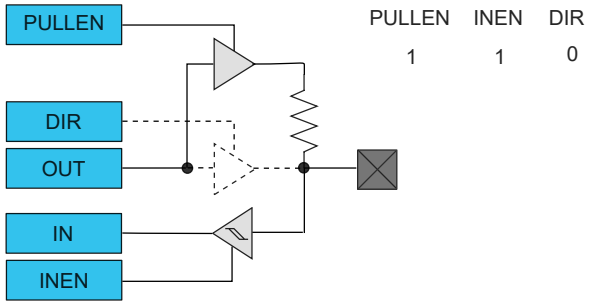


Figure 13-75. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

13.17.6.3.3 Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Figure 13-76. I/O Configuration - Totem-Pole Output with Disabled Input

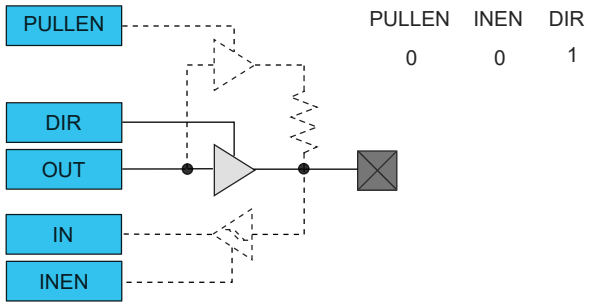


Figure 13-77. I/O Configuration - Totem-Pole Output with Enabled Input

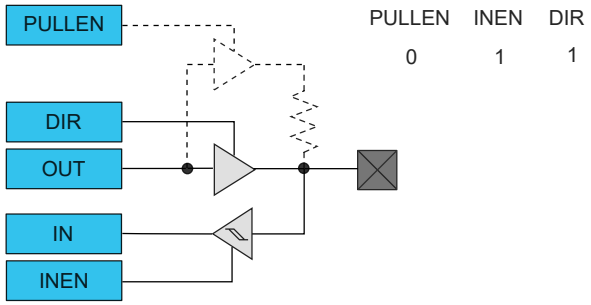
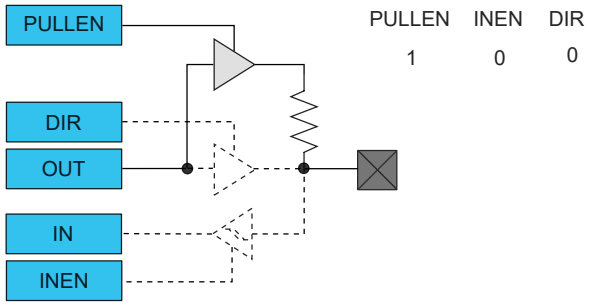


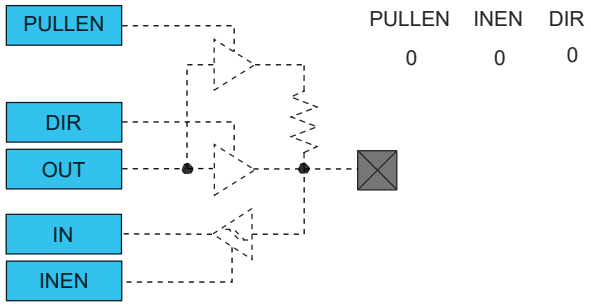
Figure 13-78. I/O Configuration - Output with Pull



13.17.6.3.4 Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 13-79. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



13.17.6.4 Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to *EVSYS – Event System*. for details on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Table 13-48. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET

.....continued				
EVACT0	**EVACT1**	**EVACT2**	**EVACT3**	**Executed Event Action**
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

Related Links
13.18 EVSYS – Event System

13.17.6.5 PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the ARM® single-cycle I/O port (IOBUS)
- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The following priority is adopted:

1. ARM® CPU IOBUS (No wait tolerated)
2. APB
3. EVSYS input events

For input events that require different actions on the same I/O pin, refer to 13.17.6.4 Events.

13.17.7 Register Summary

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DIR	7:0	DIR[7:0]							
		15:8	DIR[15:8]							
		23:16	DIR[23:16]							
		31:24	DIR[31:24]							
0x04	DIRCLR	7:0	DIRCLR[7:0]							
		15:8	DIRCLR[15:8]							
		23:16	DIRCLR[23:16]							
		31:24	DIRCLR[31:24]							
0x08	DIRSET	7:0	DIRSET[7:0]							
		15:8	DIRSET[15:8]							
		23:16	DIRSET[23:16]							
		31:24	DIRSET[31:24]							
0x0C	DIRTGL	7:0	DIRTGL[7:0]							
		15:8	DIRTGL[15:8]							
		23:16	DIRTGL[23:16]							
		31:24	DIRTGL[31:24]							
0x10	OUT	7:0	OUT[7:0]							
		15:8	OUT[15:8]							
		23:16	OUT[23:16]							
		31:24	OUT[31:24]							
0x14	OUTCLR	7:0	OUTCLR[7:0]							
		15:8	OUTCLR[15:8]							
		23:16	OUTCLR[23:16]							
		31:24	OUTCLR[31:24]							
0x18	OUTSET	7:0	OUTSET[7:0]							
		15:8	OUTSET[15:8]							
		23:16	OUTSET[23:16]							
		31:24	OUTSET[31:24]							
0x1C	OUTTGL	7:0	OUTTGL[7:0]							
		15:8	OUTTGL[15:8]							
		23:16	OUTTGL[23:16]							
		31:24	OUTTGL[31:24]							
0x20	IN	7:0	IN[7:0]							
		15:8	IN[15:8]							
		23:16	IN[23:16]							
		31:24	IN[31:24]							
0x24	CTRL	7:0	SAMPLING[7:0]							
		15:8	SAMPLING[15:8]							
		23:16	SAMPLING[23:16]							
		31:24	SAMPLING[31:24]							
0x28	WRCONFIG	7:0	PINMASK[7:0]							
		15:8	PINMASK[15:8]							
		23:16		DRVSTR				PULLEN	INEN	PMUXEN
		31:24	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
0x2C	EVCTRL	7:0	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		15:8	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		23:16	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
		31:24	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
...										
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0		DRVSTR				PULLEN	INEN	PMUXEN
...										
0x5F	PINCFG31	7:0		DRVSTR				PULLEN	INEN	PMUXEN

13.17.8 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.17.5.8 Register Access Protection.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

13.17.8.1 Data Direction

Name: DIR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0] Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as an input.
1	The corresponding I/O pin in the PORT group is configured as an output.

13.17.8.2 Data Direction Clear

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0] Port Data Direction Clear
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

13.17.8.3 Data Direction Set

Name: DIRSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0] Port Data Direction Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

13.17.8.4 Data Direction Toggle

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0] Port Data Direction Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

13.17.8.5 Data Output Value

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

This register sets the data output drive value for the individual I/O pins in the PORT.
This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0] Port Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.
For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

13.17.8.6 Data Output Value Clear

Name: OUTCLR
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0] PORT Data Output Value Clear
Writing '0' to a bit has no effect.
Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

13.17.8.7 Data Output Value Set

Name: OUTSET
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0] PORT Data Output Value Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up.

13.17.8.8 Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.
The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0] PORT Data Output Value Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding OUT bit value is toggled.

13.17.8.9 Data Input Value

Name: IN
Offset: 0x20
Reset: 0x00000000
Property: -

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0] PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.
These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

13.17.8.10 Control

Name: CTRL
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SAMPLING[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SAMPLING[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0] Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).
The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Value	Description
0	The I/O pin input synchronizer is disabled.
1	The I/O pin input synchronizer is enabled.

13.17.8.11 Write Configuration

Name: WRCONFIG
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

This write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W	W		W	W	W	W	W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DRVSTR				PULLEN	INEN	PMUXEN
Access		W				W	W	W
Reset		0				0	0	0

Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINMASK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL Half-Word Select
This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.
This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFG Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.
Writing '0' to this bit has no effect.
Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN and WRCONFIG.PINMASK values.
This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.
Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.
This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0] Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.
These bits will always read as zero.

Bit 22 – DRVSTR Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 18 – PULLEN Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 17 – INEN Input Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 16 – PMUXEN Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bits 15:0 – PINMASK[15:0] Pin Mask for Multiple Pin Configuration
These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.
These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word PORT group will be updated.

13.17.8.12 Event Input Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x00000000
Property: PAC Write-Protection

The I/O pins are assembled in PORT groups with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each PORT group has its own set of PORT registers with offset 0x80. The available number of PORT groups may depend on the package/pin number of the device.

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Bit	31	30	29	28	27	26	25	24
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PORTEIx	EVACTx[1:0]		PIDx[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31,23,15,7 – PORTEIx PORT Event Input Enable x [x = 3..0]

Value	Description
0	The event action x (EVACTx) will not be triggered on any incoming event.
1	The event action x (EVACTx) will be triggered on any incoming event.

Bits 30:29, 22:21,14:13,6:5 – EVACTx PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 13-49.

Bits 28:24,20:16,12:8,4:0 – PIDx PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 13-50.

Table 13-49. PORT Event x Action (x = [3..0])

Value	Name	Description
0x0	OUT	Output register of pin will be set to level of event.
0x1	SET	Set output register of pin on event.
0x2	CLR	Clear output register of pin on event.
0x3	TGL	Toggle output register of pin on event.

Table 13-50. PORT Event x Pin Identifier (x = [3..0])

Value	Name	Description
0x0	PIN0	Event action to be executed on PIN 0.
0x1	PIN1	Event action to be executed on PIN 1.
...	...	...
0x31	PIN31	Event action to be executed on PIN 31.

13.17.8.13 Peripheral Multiplexing n

Name: PMUX
Offset: 0x30 + n*0x01 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines.

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0] Peripheral Multiplexing for Odd-Numbered Pin
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

PMUXO[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0] Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

PMUXE[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected

.....continued		
PMUXE[3:0]	Name	Description
0x9-0xF	-	Reserved

Related Links

7. I/O Multiplexing and Considerations

13.17.8.14 Pin Configuration

Name: PINCFG
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: PAC Write-Protection

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 6 – DRVSTR Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

Bit 2 – PULLEN Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled, and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.
Writing a zero to this bit disables the input buffer completely, preventing read-back of the physical pin state when the pin is configured as either an input or output.

Value	Description
0	Input buffer for the I/O pin is disabled, and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled, and the input value will be sampled when required.

Bit 0 – PMUXEN Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.
Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output drive value via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the physical pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value.

13.18 EVSYS – Event System

13.18.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

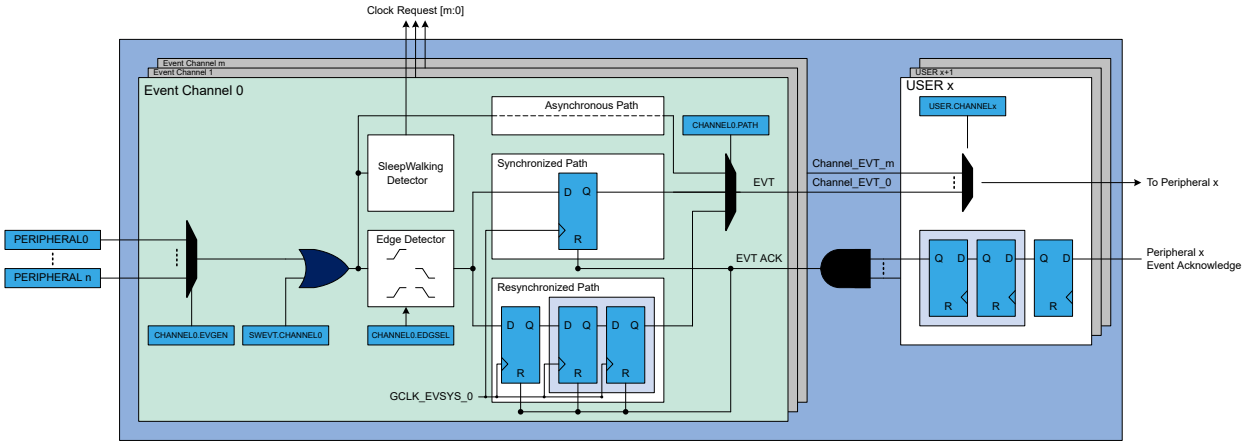
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

13.18.2 Features

- 12 configurable event channels, where each channel can:
  - Be connected to any event generator.
  - Provide a pure asynchronous, resynchronized or synchronous path
- 82 event generators.
- 42 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.

13.18.3 Block Diagram

Figure 13-80. Event System Block Diagram



13.18.4 Signal Description

Not applicable.

13.18.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.18.5.1 I/O Lines

Not applicable.

13.18.5.2 Power Management

The EVSYS can be used to wake up the CPU from all sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

In all sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Related Links

13.8 PM – Power Manager

13.18.5.3 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.5 GCLK - Generic Clock Controller

13.18.5.4 DMA

Not applicable.

13.18.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.18.5.6 Events

Not applicable.

13.18.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.18.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Channel Status (CHSTATUS)
- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.18.5.9 Analog Connections

Not applicable.

13.18.6 Functional Description

13.18.6.1 Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or IO pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or re-synchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on on event generators.

For further details, refer to "13.18.6.2.6 Channel Path" of this chapter.

13.18.6.2 Basic Operation

13.18.6.2.1 Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event have to be configured. The recommended sequence is:

1. In the event generator peripheral, enable output of event by writing a '1' to the respective Event Output Enable bit ("EO") in the peripheral's Event Control register (e.g., TCC.EVCTRL.MCEO1, AC.EVCTRL.WINEO0, RTC.EVCTRL.OVFEO).
2. Configure the EVSYS:
   2.1. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, see also 13.18.6.2.3 User Multiplexer Setup.
   2.2. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, see also 13.18.6.2.4 Event System Channel.
3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register (e.g., TC.EVCTRL.EVACT, PDEC.EVCTRL.EVACT). Note: not all peripherals require this step.
4. In the event user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register (e.g., AC.EVCTRL.IVEI0, ADC.EVCTRL.STARTEI).

13.18.6.2.2 Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to 13.18.8.1 CTRLA.SWRST register for details.

13.18.6.2.3 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Figure 13-80. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

13.18.6.2.4 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in Figure 13-80.

13.18.6.2.5 Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

13.18.6.2.6 Channel Path

There are three different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel Status register (CHSTATUS) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Synchronous Path

The synchronous path should be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When using the synchronous path, the channel is able to generate interrupts. The channel busy n bit in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When the resynchronized path is used, the channel is able to generate interrupts. The channel busy n bits in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Related Links

13.5 GCLK - Generic Clock Controller

13.18.6.2.7 Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

13.18.6.2.8 Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- Asynchronous Path: The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.

- Synchronous Path: The maximum routing latency of an external event is one GCLK_EVSYS_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

13.18.6.2.9 The Overrun Channel n Interrupt

The Overrun Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVRn) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event.
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel.

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.OVRn is always read as zero.

13.18.6.2.10 The Event Detected Channel n Interrupt

The Event Detected Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.EVDn) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.EVDn is always zero.

13.18.6.2.11 Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUS.CHBUSYn bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUS.USRRDYn bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

13.18.6.2.12 Software Event

A software event can be initiated on a channel by setting the Channel n bit in the Software Event register (SWEVT.CHANNELn) to '1'. Then the software event can be serviced as any event generator; i.e., when the bit is set to '1', an event will be generated on the respective channel.

13.18.6.3 Interrupts

The EVSYS has the following interrupt sources:

- Overrun Channel n interrupt (OVRn): For details, refer to 13.18.6.2.9 The Overrun Channel n Interrupt.
- Event Detected Channel n interrupt (EVDn): For details, refer to 13.18.6.2.10 The Event Detected Channel n Interrupt.

These interrupt events are asynchronous wake-up sources. See *Sleep Mode Controller*. Each interrupt source has an interrupt flag which is in the Interrupt Flag Status and Clear (INTFLAG) register. The flag is set when the interrupt is issued. Each interrupt event can be individually enabled by setting a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by setting a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt event is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt event works until the interrupt flag is cleared, the interrupt is disabled or the Event System is reset. See 13.18.8.5 INTFLAG for details on how to clear interrupt flags.

All interrupt events from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the *Nested Vector Interrupt Controller* for details. The event user must read the INTFLAG register to determine what the interrupt condition is.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

13.18.6.4 Sleep Mode Operation

The EVSYS can generate interrupts to wake up the device from any sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register (CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND, as shown in the table below:

Table 13-51. Event Channel Sleep Behavior

CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
0	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode.
0	1	Always run in IDLE and STANDBY sleep modes.
1	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.
1	1	Always run in IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

13.18.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x0B	Reserved									
0x0C	CHSTATUS	7:0	USRRDY7	USRRDY6	USRRDY5	USRRDY4	USRRDY3	USRRDY2	USRRDY1	USRRDY0
		15:8					USRRDY11	USRRDY10	USRRDY9	USRRDY8
		23:16	CHBUSY7	CHBUSY6	CHBUSY5	CHBUSY4	CHBUSY3	CHBUSY2	CHBUSY1	CHBUSY0
		31:24					CHBUSY11	CHBUSY10	CHBUSY9	CHBUSY8
0x10	INTENCLR	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
		15:8					OVR11	OVR10	OVR9	OVR8
		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
		31:24					EVD11	EVD10	EVD9	EVD8
0x14	INTENSET	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
		15:8					OVR11	OVR10	OVR9	OVR8
		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
		31:24					EVD11	EVD10	EVD9	EVD8
0x18	INTFLAG	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
		15:8					OVR11	OVR10	OVR9	OVR8
		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
		31:24					EVD11	EVD10	EVD9	EVD8
0x1C	SWEVT	7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
		15:8					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		23:16								
		31:24								
0x20	CHANNEL0	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
...										
0x4C	CHANNEL11	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x50 ... 0x7F	Reserved									
0x80	USER0	7:0			CHANNEL[5:0]					
		15:8								
		23:16								
		31:24								
...										
0x0124	USER41	7:0			CHANNEL[5:0]					
		15:8								
		23:16								
		31:24								

13.18.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to 13.18.5.8 Register Access Protection and *PAC - Peripheral Access Controller*.

13.18.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								0

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state. It will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Note: Before applying a Software Reset it is recommended to disable the event generators.

13.18.8.2 Channel Status

Name: CHSTATUS
Offset: 0x0C
Reset: 0x000000FF
Property: –

Bit	31	30	29	28	27	26	25	24
					CHBUSY11	CHBUSY10	CHBUSY9	CHBUSY8
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHBUSY7	CHBUSY6	CHBUSY5	CHBUSY4	CHBUSY3	CHBUSY2	CHBUSY1	CHBUSY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					USRRDY11	USRRDY10	USRRDY9	USRRDY8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USRRDY7	USRRDY6	USRRDY5	USRRDY4	USRRDY3	USRRDY2	USRRDY1	USRRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 – CHBUSYn Channel Busy n [n = 11..0]
This bit is cleared when channel n is idle.
This bit is set if an event on channel n has not been handled by all event users connected to channel n.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – USRRDYn User Ready for Channel n [n = 11..0]
This bit is cleared when at least one of the event users connected to the channel is not ready.
This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.

13.18.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 – EVDn Event Detected Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Event Detected Channel n Interrupt Enable bit, which disables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – OVRn Overrun Channel n Interrupt Enable[n = 11..0]
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

13.18.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 – EVDn Event Detected Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Event Detected Channel n Interrupt Enable bit, which enables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – OVRn Overrun Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

13.18.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 – EVDn Event Detected Channel n [n=11..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.EVDn is '1'.
When the event channel path is asynchronous, the EVDn interrupt flag will not be set.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Event Detected Channel n interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – OVRn Overrun Channel n [n=11..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.OVRn is '1'.
When the event channel path is asynchronous, the OVRn interrupt flag will not be set.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Overrun Detected Channel n interrupt flag.

13.18.8.6 Software Event

Name: SWEVT
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHANNELn Channel n Software [n=11..0] Selection
Writing '0' to this bit has no effect.
Writing '1' to this bit will trigger a software event for the channel n.
These bits will always return zero when read.

13.18.8.7 Channel

Name: CHANNELn
Offset: 0x20 + n*0x04 [n=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to configure channel n. To write to this register, do a single 32-bit write of all the configuration data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
		EVGEN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 15 – ONDEMAND Generic Clock On Demand
This bit is used to determine whether the generic clock is requested.

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in Standby
This bit is used to define the behavior during standby sleep mode.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection
These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0] Path Selection
These bits are used to choose which path will be used by the selected channel.
The path choice can be limited by the channel source; see the table in 13.18.8.8 USERm.

Value	Name	Description
0x0	SYNCHRONOUS	Synchronous path

Value	Name	Description
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
0x3	—	Reserved

Bits 6:0 – EVGEN[6:0] Event Generator
These bits are used to choose the event generator to connect to the selected channel.

Value	Event Generator	Description
0x00	NONE	No event generator selected
0x01	RTC CMP0	Compare 0 (mode 0 and 1) or Alarm 0 (mode 2)
0x02	RTC CMP1	Compare 1
0x03	RTC OVF	Overflow
0x04	RTC PER0	Period 0
0x05	RTC PER1	Period 1
0x06	RTC PER2	Period 2
0x07	RTC PER3	Period 3
0x08	RTC PER4	Period 4
0x09	RTC PER5	Period 5
0x0A	RTC PER6	Period 6
0x0B	RTC PER7	Period 7
0x0C	EIC EXTINT0	External Interrupt 0
0x0D	EIC EXTINT1	External Interrupt 1
0x0E	EIC EXTINT2	External Interrupt 2
0x0F	EIC EXTINT3	External Interrupt 3
0x10	EIC EXTINT4	External Interrupt 4
0x11	EIC EXTINT5	External Interrupt 5
0x12	EIC EXTINT6	External Interrupt 6
0x13	EIC EXTINT7	External Interrupt 7
0x14	EIC EXTINT8	External Interrupt 8
0x15	EIC EXTINT9	External Interrupt 9
0x16	EIC EXTINT10	External Interrupt 10
0x17	EIC EXTINT11	External Interrupt 11
0x18	EIC EXTINT12	External Interrupt 12
0x19	EIC EXTINT13	External Interrupt 13
0x1A	EIC EXTINT14	External Interrupt 14
0x1B	EIC EXTINT15	External Interrupt 15
0x1C	DMAC CH0	Channel 0
0x1D	DMAC CH1	Channel 1
0x1E	DMAC CH2	Channel 2
0x1F	DMAC CH3	Channel 3
0x20	DMAC CH4	Channel 4
0x21	DMAC CH5	Channel 5
0x22	DMAC CH6	Channel 6
0x23	DMAC CH7	Channel 7
0x24	TCC0 OVF	Overflow
0x25	TCC0 TRG	Trig
0x26	TCC0 CNT	Counter
0x27	TCC0_MCX0	Match/Capture 0
0x28	TCC0_MCX1	Match/Capture 1
0x29	TCC0_MCX2	Match/Capture 2
0x2A	TCC0_MCX3	Match/Capture 3
0x2B	TCC1 OVF	Overflow
0x2C	TCC1 TRG	Trig
0x2D	TCC1 CNT	Counter

.....continued

Value	Event Generator	Description
0x2E	TCC1_MCX0	Match/Capture 0
0x2F	TCC1_MCX1	Match/Capture 1
0x30	TCC2 OVF	Overflow
0x31	TCC2 TRG	Trig
0x32	TCC2 CNT	Counter
0x33	TCC2_MCX0	Match/Capture 0
0x34	TCC2_MCX1	Match/Capture 1
0x35	TC0 OVF	Overflow/Underflow
0x36	TC0 MC0	Match/Capture 0
0x37	TC0 MC1	Match/Capture 1
0x38	TC1 OVF	Overflow/Underflow
0x39	TC1 MC0	Match/Capture 0
0x3A	TC1 MC1	Match/Capture 1
0x3B	Reserved	Reserved
0x3C	Reserved	Reserved
0x3D	Reserved	Reserved
0x3E	Reserved	Reserved
0x3F	Reserved	Reserved
0x40	Reserved	Reserved
0x41	TC4 OVF	Overflow/Underflow
0x42	TC4 MC0	Match/Capture 0
0x43	TC4 MC1	Match/Capture 1
0x44	ADC RESRDY	Result Ready
0x45	ADC WINMON	Window Monitor
0x46	AC COMP0	Comparator 0
0x47	AC COMP1	Comparator 1
0x48	AC WIN0	Window 0
0x49	Reserved	Reserved
0x4A	Reserved	Reserved
0x4B	PTC EOC	End of Conversion
0x4C	PTC WCOMP	Window Comparator
0x4D	Reserved	Reserved
0x4E	CCL LUTOUT0	CCL output
0x4F	CCL LUTOUT1	CCL output
0x50	CCL LUTOUT2	CCL output
0x51	CCL LUTOUT3	CCL output
0x52	PAC ACCERR	Access Error
0x53-0x7F	Reserved	—

13.18.8.8 Event User m

Name: USERm
Offset: 0x80 + m*0x04 [m=0..41]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CHANNEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – CHANNEL[5:0] Channel Event Selection
These bits are used to select the channel to connect to the event user.
Note that to select channel m, the value (m+1) must be written to the USER.CHANNEL bit group.

Value	Channel Number
0x00	No channel output selected
0x01	0
0x02	1
0x03	2
0x04	3
0x05	4
0x06	5
0x07	6
0x08	7
0x09	8
0x0A	9
0x0B	10
0x0C	11
0x0D-0xFF	Reserved

Table 13-52. User Multiplexer Number

USERm	User Multiplexer	Description	Path Type
m = 0	PORT EV0	Event 0	Asynchronous, synchronous and resynchronized paths
m = 1	PORT EV1	Event 1	Asynchronous, synchronous and resynchronized paths

.....continued

USERm	User Multiplexer	Description	Path Type
m = 2	PORT EV2	Event 2	Asynchronous, synchronous and resynchronized paths
m = 3	PORT EV3	Event 3	Asynchronous, synchronous and resynchronized paths
m = 4	DMAC CH0	Channel 0	Synchronous and resynchronized paths
m = 5	DMAC CH1	Channel 1	Synchronous and resynchronized paths
m = 6	DMAC CH2	Channel 2	Synchronous and resynchronized paths
m = 7	DMAC CH3	Channel 3	Synchronous and resynchronized paths
m = 8	DMAC CH4	Channel 4	Synchronous and resynchronized paths
m = 9	DMAC CH5	Channel 5	Synchronous and resynchronized paths
m = 10	DMAC CH6	Channel 6	Synchronous and resynchronized paths
m = 11	DMAC CH7	Channel 7	Synchronous and resynchronized paths
m = 12	TCC0 EV0	—	Asynchronous, synchronous and resynchronized paths
m = 13	TCC0 EV1	—	Asynchronous, synchronous and resynchronized paths
m = 14	TCC0 MC0	Match/Capture 0	Asynchronous, synchronous and resynchronized paths
m = 15	TCC0 MC1	Match/Capture 1	Asynchronous, synchronous and resynchronized paths
m = 16	TCC0 MC2	Match/Capture 2	Asynchronous, synchronous and resynchronized paths
m = 17	TCC0 MC3	Match/Capture 3	Asynchronous, synchronous and resynchronized paths
m = 18	TCC1 EV0	—	Asynchronous, synchronous and resynchronized paths
m = 19	TCC1 EV1	—	Asynchronous, synchronous and resynchronized paths
m = 20	TCC1 MC0	Match/Capture 0	Asynchronous, synchronous and resynchronized paths
m = 21	TCC1 MC1	Match/Capture 1	Asynchronous, synchronous and resynchronized paths
m = 22	TCC2 EV0	—	Asynchronous, synchronous and resynchronized paths

.....continued

USERm	User Multiplexer	Description	Path Type
m = 23	TCC2 EV1	—	Asynchronous, synchronous and resynchronized paths
m = 24	TCC2 MC0	Match/Capture 0	Asynchronous, synchronous and resynchronized paths
m = 25	TCC2 MC1	Match/Capture 1	Asynchronous, synchronous and resynchronized paths
m = 26	TC0	—	Asynchronous, synchronous and resynchronized paths
m = 27	TC1	—	Asynchronous, synchronous and resynchronized paths
m = 28	Reserved	—	—
m = 29	Reserved	—	—
m = 30	TC4	—	Asynchronous, synchronous and resynchronized paths
m = 31	ADC START	ADC start conversion	Asynchronous, synchronous and resynchronized paths
m = 32	ADC SYNC	Flush ADC	Asynchronous, synchronous and resynchronized paths
m = 33	AC COMP0	Start comparator 0	Asynchronous, synchronous and resynchronized paths
m = 34	AC COMP1	Start comparator 1	Asynchronous, synchronous and resynchronized paths
m = 35	Reserved	—	—
m = 36	Reserved	—	—
m = 37	PTC STCONV	PTC start conversion	Asynchronous, synchronous and resynchronized paths
m = 38	CCL LUTIN 0	CCL input	Asynchronous, synchronous and resynchronized paths
m = 39	CCL LUTIN 1	CCL input	Asynchronous, synchronous and resynchronized paths
m = 40	CCL LUTIN 2	CCL input	Asynchronous, synchronous and resynchronized paths
m = 41	CCL LUTIN 3	CCL input	Asynchronous, synchronous and resynchronized paths
m = 42	Reserved	—	—
m = 43	MTB START	Tracing start	Asynchronous, synchronous and resynchronized paths

.....continued			
USERm	**User Multiplexer**	**Description**	**Path Type**
m = 44	MTB STOP	Tracing stop	Asynchronous, synchronous and resynchronized paths
others	Reserved	—	—

13.19 SERCOM – Serial Communication Interface

13.19.1 Overview

There are up to six instances of the serial communication interface (SERCOM) peripheral. Up to five (SERCOM[4:0]) are located in PD1, whereas SERCOM5, present in all device configurations, is always located in power domain PD0.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When SERCOM is configured and enabled, all SERCOM resources will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock to operate in all sleep modes.

Related Links

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

13.21 SERCOM SPI – SERCOM Serial Peripheral Interface

13.22 SERCOM I2C – SERCOM Inter-Integrated Circuit

13.19.2 Features

- Interface for configuring into one of the following:
  - I^2C – Two-wire serial interface
    SMBus™ compatible
  - SPI – Serial peripheral interface
  - USART – Universal synchronous and asynchronous serial receiver and transmitter
- Single transmit buffer and double receive buffer
- Baud-rate generator
- Address match/mask logic
- Operational in all sleep modes
- Can be used with DMA

Notes: SERCOM5, due to its location in PD0, has a reduced feature set and does *not* support these features:

- General: DMA support
- USART:
  - 3x or 8x oversampling
  - Flow control (RTS/CTS)
  - IrDA
  - Single wire UART according to EN54
  - SOF/EOF function
- I^2C:
  - Fm+ and Hs modes
  - SMBus SCL low timeout
  - 10-bit addressing
  - PMBus Group command support
- SPI:
  - Hardware chip select
  - Wake on $\overline{SS}$ assertion

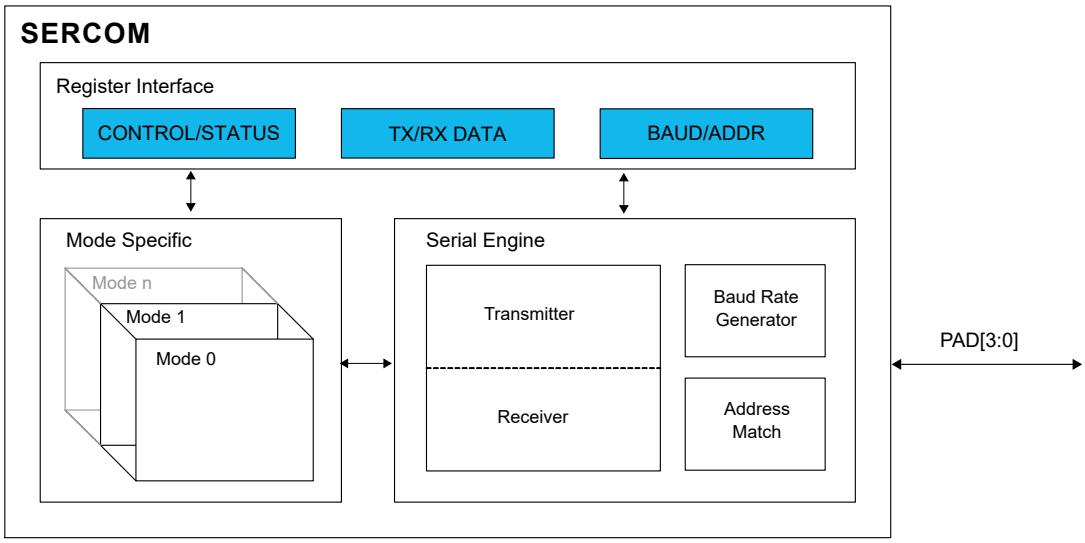
See the Related Links for full feature lists of the interface configurations.

Related Links

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter
13.21 SERCOM SPI – SERCOM Serial Peripheral Interface
13.22 SERCOM I2C – SERCOM Inter-Integrated Circuit

13.19.3 Block Diagram

Figure 13-81. SERCOM Block Diagram



13.19.4 Signal Description

See the respective SERCOM mode chapters for details.

Related Links

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter
13.21 SERCOM SPI – SERCOM Serial Peripheral Interface
13.22 SERCOM I2C – SERCOM Inter-Integrated Circuit

13.19.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.19.5.1 I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT).

From *USART Block Diagram* one can see that the SERCOM has four internal pads, PAD[3:0]. The signals from I2C, SPI and USART are routed through these SERCOM pads via a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for details.

Related Links

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter
13.21 SERCOM SPI – SERCOM Serial Peripheral Interface
13.22 SERCOM I2C – SERCOM Inter-Integrated Circuit
13.17 PORT - I/O Pin Controller
13.20.3 Block Diagram

13.19.5.2 Power Management

The SERCOM can operate in any sleep mode where the selected clock source is running. SERCOM interrupts can be used to wake up the device from sleep modes.

Related Links

13.8 PM – Power Manager

13.19.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

The SERCOM uses two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a master. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to 13.19.6.8 Synchronization for details.

Related Links

13.5 GCLK - Generic Clock Controller

13.6 MCLK – Main Clock

13.19.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.19.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

Related Links

11.2 Nested Vector Interrupt Controller

13.19.5.6 Events

Not applicable.

13.19.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.19.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.19.5.9 Analog Connections

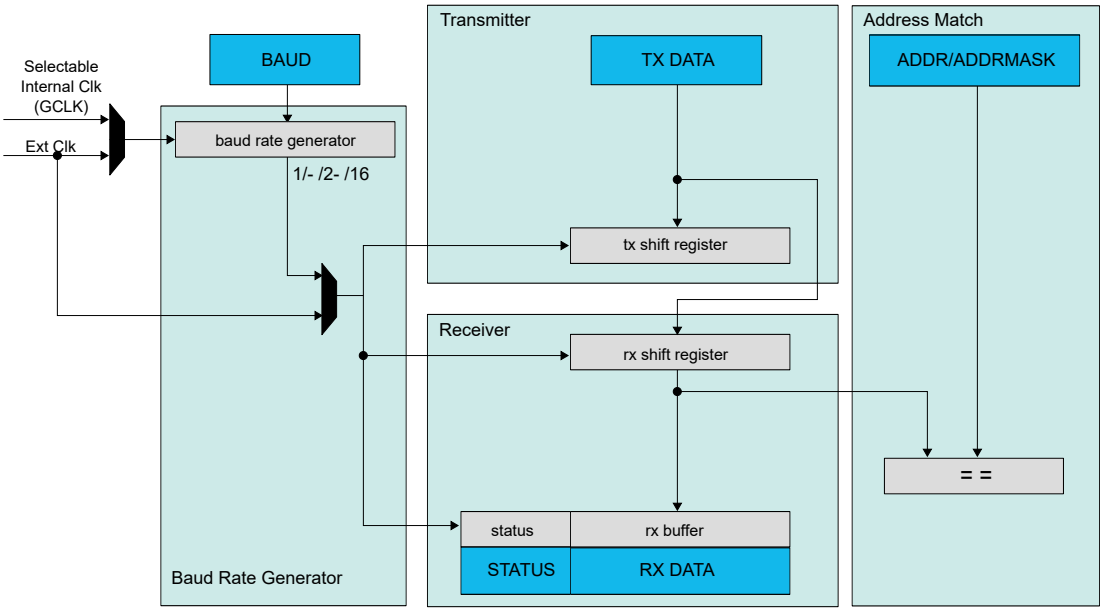
Not applicable.

13.19.6 Functional Description

13.19.6.1 Principle of Operation

The basic structure of the SERCOM serial engine is shown in Figure 13-82. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

Figure 13-82. SERCOM Serial Engine



The transmitter consists of a single write buffer and a shift register.

The receiver consists of a two-level receive buffer and a shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

13.19.6.2 Basic Operation

13.19.6.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE). Refer to table SERCOM Modes for details.

Table 13-53. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in slave operation
0x3	SPI in master operation
0x4	I^2C slave operation
0x5	I^2C master operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

Related Links

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter
13.21 SERCOM SPI – SERCOM Serial Peripheral Interface
13.22 SERCOM I2C – SERCOM Inter-Integrated Circuit

13.19.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

13.19.6.2.3 Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 13-83, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 13-83. Baud Rate Generator

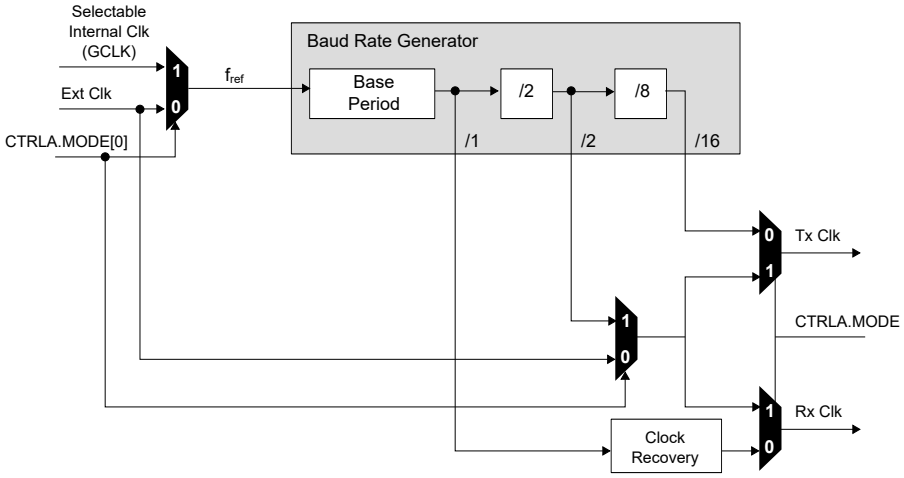


Table 13-54 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there are two different modes: In *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535). In *fractional mode*, the BAUD register is 13 bits, while the fractional adjustment is 3 bits. In this mode the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 13-54. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$

.....continued			
Operating Mode	**Condition**	**Baud Rate (Bits Per Second)**	**BAUD Register Value Calculation**
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit. Can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 13-55 shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 13-55. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3MHz
407 – 808	161	2.981MHz
809 – 1205	162	2.963MHz
...	...	...
65206	31775	15.11kHz
65207	31871	15.06kHz
65208	31969	15.01kHz

13.19.6.3 Additional Features

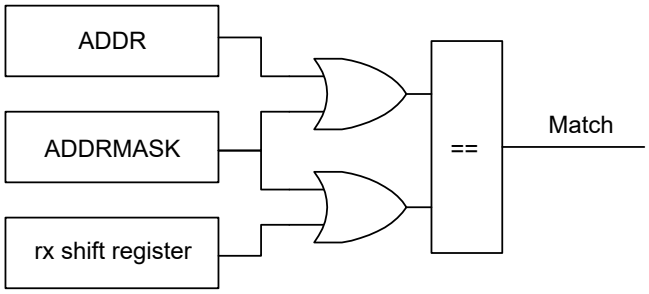
13.19.6.3.1 Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

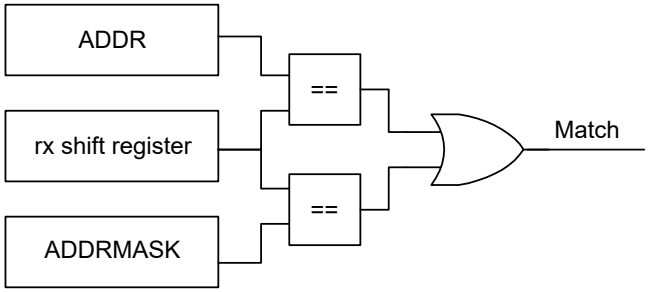
Figure 13-84. Address With Mask



Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

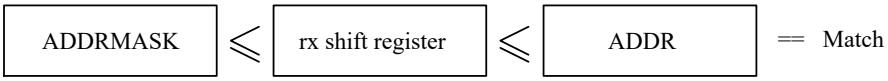
Figure 13-85. Two Unique Addresses



Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 13-86. Address Range



13.19.6.4 DMA Operation

Not applicable.

13.19.6.5 Interrupts

Interrupt sources are mode-specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own interrupt flag.

The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SERCOM has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt condition occurred. The user must read the INTFLAG register to determine which interrupt condition is present.

Note:
Note that interrupts must be globally enabled for interrupt requests.

Related Links

11.2 Nested Vector Interrupt Controller

13.19.6.6 Events

Not applicable.

13.19.6.7 Sleep Mode Operation

The peripheral can operate in any sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

13.19.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

13.4.3 Register Synchronization

13.20 SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

13.20.1 Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, see 13.20.3 Block Diagram. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

Related Links

13.19 SERCOM – Serial Communication Interface

13.20.2 USART Features

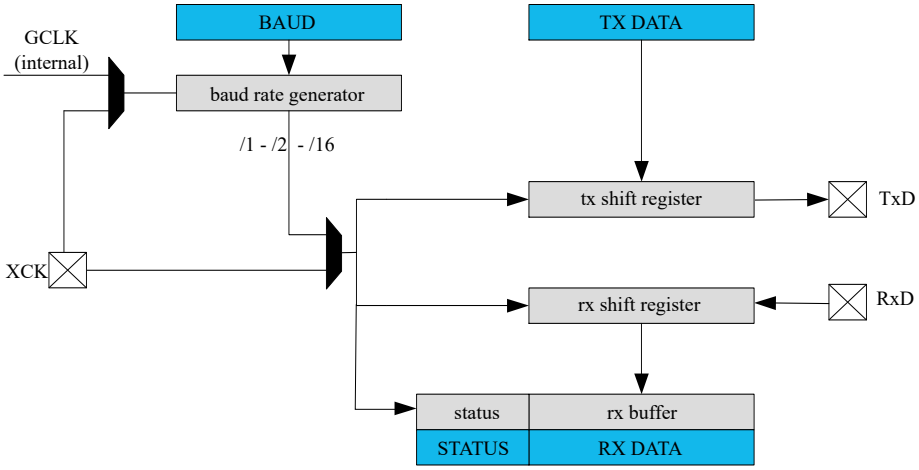
- Full-duplex operation
- Asynchronous (with clock reconstruction) or synchronous operation
- Internal or external clock source for asynchronous and synchronous operation
- Baud-rate generator
- Supports serial frames with 5, 6, 7, 8 or 9 data bits and 1 or 2 stop bits
- Odd or even parity generation and parity check
- Selectable LSB- or MSB-first data transfer
- Buffer overflow and frame error detection
- Noise filtering, including false start-bit detection and digital low-pass filter
- Collision detection
- Can operate in all sleep modes
- Operation at speeds up to half the system clock for internally generated clocks
- Operation at speeds up to the system clock for externally generated clocks
- RTS and CTS flow control
- IrDA modulation and demodulation up to 115.2kbps

- Start-of-frame detection
- Can work with DMA

Related Links
13.19.2 Features

13.20.3 Block Diagram

Figure 13-87. USART Block Diagram



13.20.4 Signal Description

Table 13-56. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links
7. I/O Multiplexing and Considerations

13.20.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.20.5.1 I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 13-57. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in Table 13-57.

Related Links

13.17 PORT - I/O Pin Controller

13.20.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.20.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be disabled and enabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the Generic Clock Controller before using the SERCOMx_CORE. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to *Synchronization* for further details.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.5 GCLK - Generic Clock Controller

13.20.6.6 Synchronization

13.20.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.20.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.20.5.6 Events

Not applicable.

13.20.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.20.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

13.20.5.9 Analog Connections

Not applicable.

13.20.6 Functional Description

13.20.6.1 Principle of Operation

The USART uses the following lines for data transfer:

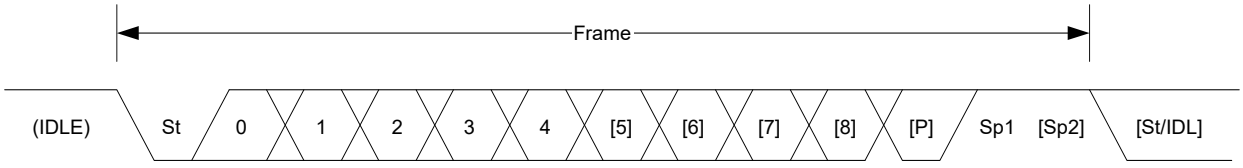
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by one character of data bits. If enabled, the parity bit is inserted after the data bits and before the first stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately, or the communication line can return to the idle (high) state. The figure below illustrates the possible frame formats. Brackets denote optional bits.

Figure 13-88. Frame Formats



St Start bit. Signal is always low.

n, [n] Data bits. 0 to [5..9]

[P] Parity bit. Either odd or even.

Sp, [Sp] Stop bit. Signal is always high.

IDLE No frame is transferred on the communication line. Signal is always high in this state.

13.20.6.2 Basic Operation

13.20.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- Baud register (BAUD)

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed. Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either Asynchronous (0) or Synchronous (1) Communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).

3. Select pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use parity mode:
   7.1. Enable Parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
   7.2. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

13.20.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

13.20.6.2.3 Clock Generation and Selection

For both synchronous and asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the asynchronous mode is selected by writing a zero to CTRLA.CMODE.

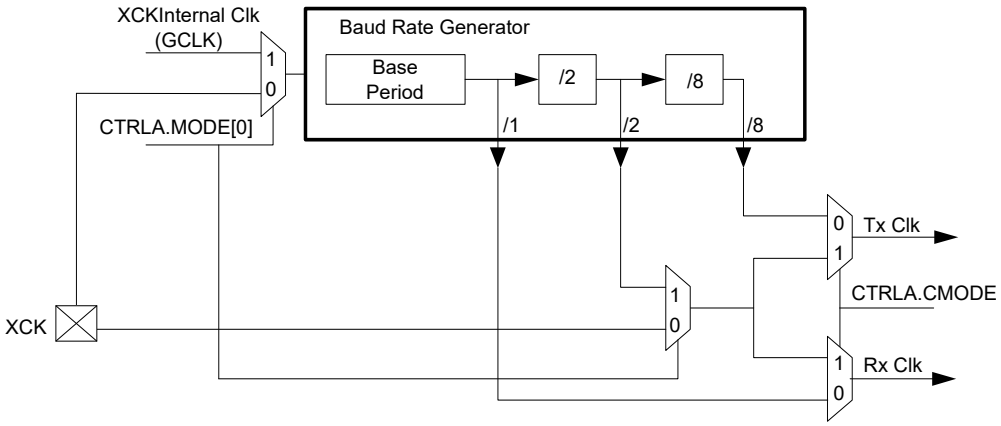
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

In asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. Refer to *Clock Generation – Baud-Rate Generator* for details on configuring the baud rate.

Figure 13-89. Clock Generation



Related Links

13.19.6.2.3 Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

Synchronous Clock Operation

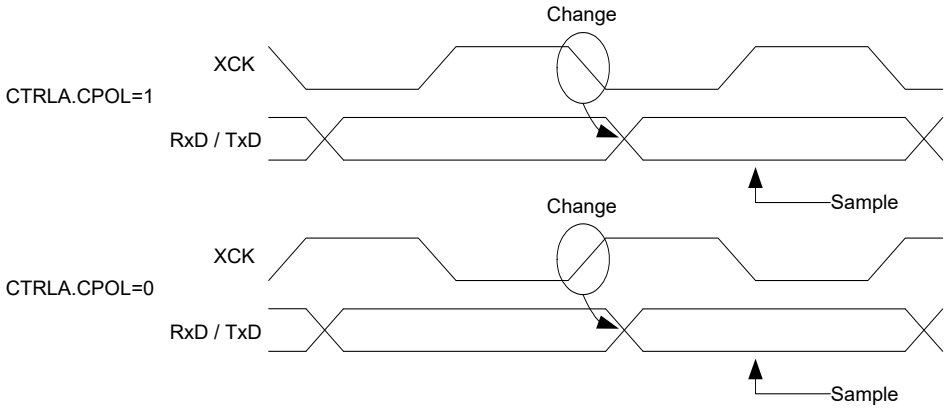
In synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.

Figure 13-90. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE=0x0), the shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

13.20.6.2.4 Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

13.20.6.2.5 Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit(s) has been transmitted and no new data was written to DATA, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the transmit shift register and TxDATA to transmit.

13.20.6.2.6 Data Reception

The receiver accepts data when a valid start bit is detected. Each bit following the start bit will be sampled according to the baud rate or XCK clock, and shifted into the receive shift register until the first stop bit of a frame is received. The second stop bit will be ignored by the receiver.

When the first stop bit is received and a complete serial frame is present in the receive shift register, the contents of the shift register will be moved into the two-level receive buffer. Then, the Receive Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set, and the optional interrupt will be generated.

The received data can be read from the DATA register when the Receive Complete interrupt flag is set.

Disabling the Receiver
Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

Error Bits
The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing ‘1’ to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

When CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA, until the receiver complete interrupt flag (INTFLAG.RXC) is cleared.

When CTRLA.IBON=0, the buffer overflow condition is attending data through the receive FIFO. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

Asynchronous Data Reception
The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

Asynchronous Operational Range
The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames, and the frame size (in number of bits). In addition, the operational range of the receiver is depending on the difference between the received bit rate and the internally generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally generated baud rate, the receiver will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have some minor instability. Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired. In this case, the BAUD register value should be set to give the lowest possible error. Refer to *Clock Generation – Baud-Rate Generator* for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

Table 13-58. Asynchronous Receiver Error for 16-fold Oversampling

D (Data bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. total error [%]	Recommended max. Rx error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

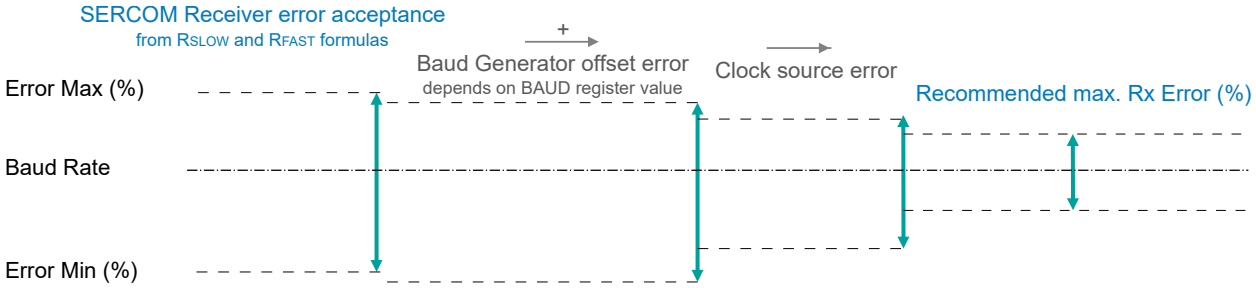
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D + 1)S}{S - 1 + D \cdot S + S_F} \quad , \qquad R_{FAST} = \frac{(D + 2)S}{(D + 1)S + S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- D is the sum of character size and parity size (D = 5 to 10 bits)
- S is the number of samples per bit (S = 16, 8 or 3)
- S_F is the first sample number used for majority voting (S_F = 7, 3, or 2) when CTRLA.SAMPA=0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4, or 2) when CTRLA.SAMPA=0.

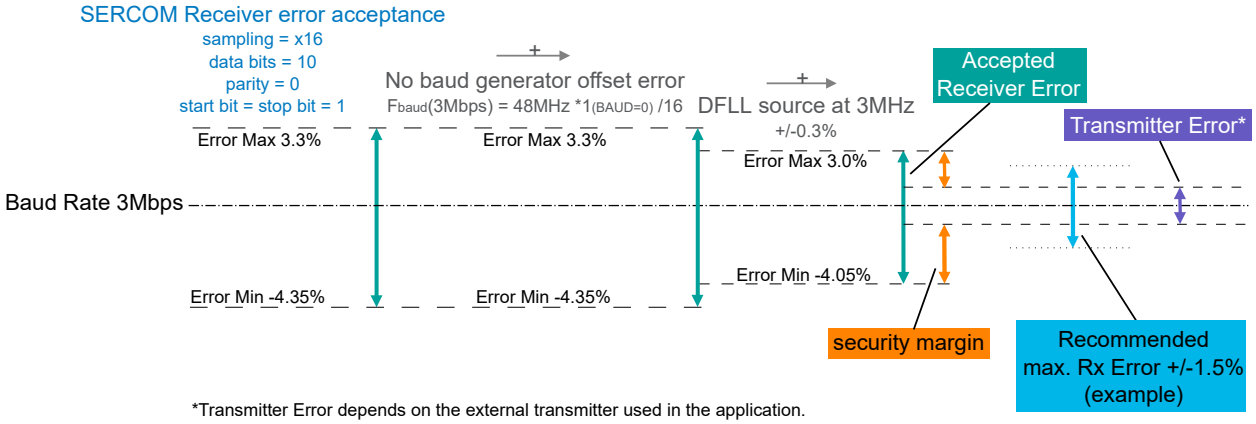
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 13-91. USART Rx Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:

Figure 13-92. USART Rx Error Calculation Example



*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

Related Links

13.19.6.2.3 Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

13.20.6.3 Additional Features

13.20.6.3.1 Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If *even parity* is selected (CTRLB.PMODE=0), the parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

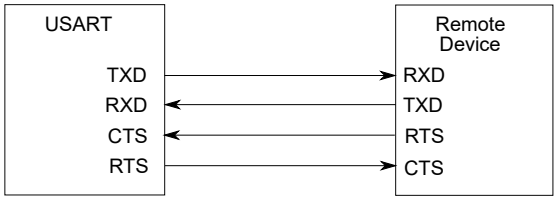
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

13.20.6.3.2 Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as shown in the figure below.

Figure 13-93. Connection with a Remote Device for Hardware Handshaking

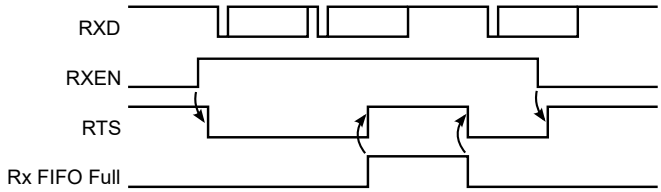


Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and Flow control pinout (CTRLA.TXPO=2).

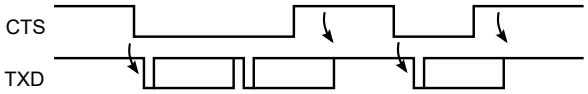
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the shift register until the receive FIFO is no longer full.

Figure 13-94. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

Figure 13-95. Transmitter Behavior when Operating with Hardware Handshaking



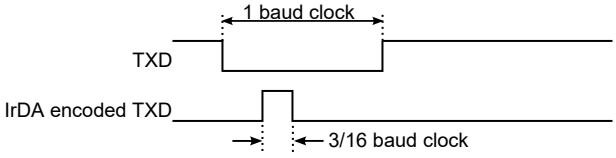
13.20.6.3.3 IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and 16x sample rate (CTRLA.SAMPR[0]=0).

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the figure below.

Figure 13-96. IrDA Transmit Encoding



The reception decoder has two main functions.

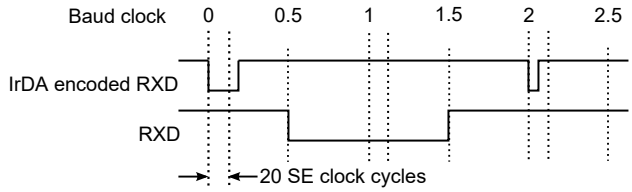
The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

The second main function is to decode incoming Rx data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

Note: Note that the polarity of the transmitter and receiver are opposite: During transmission, a '0' bit is transmitted as a '1' pulse. During reception, an accepted '0' pulse is received as a '0' bit.

Example: The figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should be at least 20 SE clock cycles. When using BAUD=0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as minimum pulse width required. In this case the first bit is accepted as a '0', the second bit is a '1', and the third bit is also a '1'. A low pulse is rejected since it does not meet the minimum requirement of 2/16 baud clock.

Figure 13-97. IrDA Receive Decoding



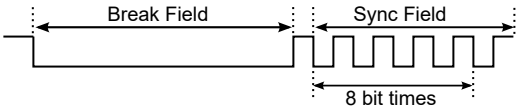
13.20.6.3.4 Break Character Detection and Auto-Baud

Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1).

The auto-baud follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field. The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a Break Field. When a Break Field has been detected, the Receive Break interrupt flag (INTFLAG.RXBRK) is set and the USART expects the Sync Field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized. If the received Sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error interrupt flag (INTFLAG.ERROR), and the baud rate is unchanged.

Figure 13-98. LIN Break and Sync Fields



After a break field is detected and the start bit of the Sync Field is detected, a counter is started. The counter is then incremented for the next 8 bit times of the Sync Field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 most significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 least significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

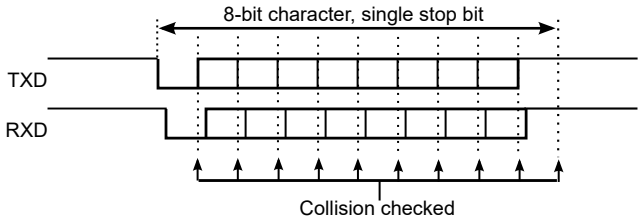
When the Sync Field has been received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the Break and Sync Fields are received, multiple characters of data can be received.

13.20.6.3.5 Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN=1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN=1 and CTRLB.TXEN=1).

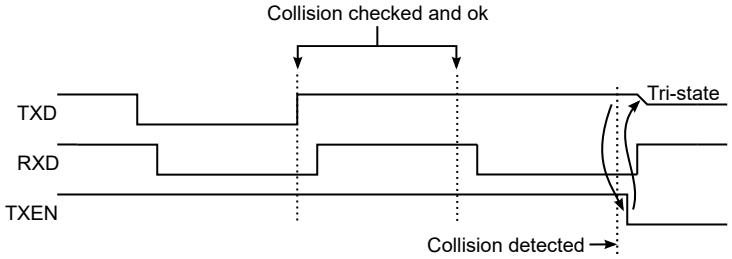
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 13-99. Collision Checking



The next figure shows the conditions for a collision detection. In this case, the start bit and the first data bit are received with the same value as transmitted. The second received data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 13-100. Collision Detected



When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error interrupt flag (INTFLAG.ERROR).
5. Set the Transmit Complete interrupt flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

13.20.6.3.6 Loop-Back Mode

For loop-back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

13.20.6.3.7 Start-of-Frame Detection

The USART start-of-frame detector can wake up the CPU when it detects a start bit. In standby sleep mode, the internal fast startup oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8MHz Internal Oscillator is powered up and the USART clock is enabled. After startup, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast startup internal oscillator start-up time. Refer to *Electrical Characteristics* for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in asynchronous and synchronous modes. It is enabled by writing ‘1’ to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8MHz Internal Oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the Receive Complete interrupt is generated.

Related Links

15. Electrical Characteristics

13.20.6.3.8 Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

13.20.6.4 DMA, Interrupts and Events

Table 13-59. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

13.20.6.4.1 DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

13.20.6.4.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The USART has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.20.6.4.3 Events

Not applicable.

13.20.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Start and the Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Start and the Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

13.20.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also 13.20.8.2 CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

13.20.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8	SAMPR[2:0]							IBON
		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
		15:8			PMODE			ENC	SFDE	COLDEN
		23:16							RXEN	TXEN
		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0			COLL	ISF	CTS	BUFOVF	FERR	PERR
		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
		15:8								DATA[8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

13.20.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.20.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]							IBON
Access	R/W	R/W	R/W					R
Reset	0	0	0					0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity
This bit selects the relationship between data output change and data input sampling in synchronous mode.
This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode
This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format
These bits define the frame format.
These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity

.....continued	
FORM[3:0]	**Description**
0x2-0x3	Reserved
0x4	Auto-baud - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6-0xF	Reserved

Bits 23:22 – SAMPA[1:0] Sample Adjustment
These bits define the sample adjustment.
These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout
These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0] Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	SERCOM PAD[2]	SERCOM PAD[3]	N/A	N/A
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3				

Bits 15:13 – SAMPR[2:0] Sample Rate
These bits select the sample rate.
These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 8 – IBON Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.

Value	Description
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby
This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device can wake up on Receive Start or Transfer Complete interrupt.
0x1	Wake on Receive Start or Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode
These bits select the USART serial communication interface of the SERCOM.
These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.20.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.
Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.
Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN Transmitter Enable
Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.
Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.
Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.TXEN will read back as '1'.
This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

Bit 13 – PMODE Parity Mode
This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value

for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.
This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

Bit 10 – ENC Encoding Format
This bit selects the data encoding format.
This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE Start of Frame Detection Enable
This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.
This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN Collision Detection Enable
This bit enables collision detection.
This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode
This bit selects the number of stop bits transmitted.
This bit is not synchronized.

Value	Description
0	One stop bit.
1	Two stop bits.

Bits 2:0 – CHSIZE[2:0] Character Size
These bits select the number of bits in a character.
These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

13.20.8.3 Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	BAUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BAUD[15:0] Baud Value
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):
These bits control the clock generation, as described in the SERCOM Baud Rate section.
If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[21:0]: Baud Value**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

Related Links
13.19.6.2.3 Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

13.20.8.4 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RXPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXPL[7:0] Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.

$PULSE \geq \left(\mathrm{RXPL} + 2\right) \cdot SE_{per}$

13.20.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

13.20.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR) .

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

13.20.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R	R/W	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 5 – RXBRK Receive Break
This flag is cleared by writing '1' to it.
This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 4 – CTSIC Clear to Send Input Change
This flag is cleared by writing a '1' to it.
This flag is set when a change is detected on the CTS pin.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 3 – RXS Receive Start
This flag is cleared by writing '1' to it.
This flag is set when a start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Start interrupt flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.
This flag is set when there are unread data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
This flag is set when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready to be written.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

13.20.8.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			COLL	ISF	CTS	BUFOVF	FERR	PERR
Access			R/W	R/W	R	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – COLL Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Bit 4 – ISF Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Bit 3 – CTS Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Bit 1 – FERR Frame Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Bit 0 – PERR Parity Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5) and a parity error is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

13.20.8.9 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.
Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.
Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

13.20.8.10 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
								DATA[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0] Data
Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.
Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

13.20.8.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

13.21 SERCOM SPI – SERCOM Serial Peripheral Interface

13.21.1 Overview

The serial peripheral interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in 13.21.3 Block Diagram. Each side, master and slave, depicts a separate SPI containing a shift register, a transmit buffer and two receive buffers. In addition, the SPI master uses the SERCOM baud-rate generator, while the SPI slave can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

Related Links
13.19 SERCOM – Serial Communication Interface

13.21.2 Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- Single-buffered transmitter, double-buffered receiver
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB- or MSB-first data transfer
- Can be used with DMA
- Master operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
- Slave operation:
  - Serial clock speed, $f_{SCK}=1/t_{SSCK}$(1)
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

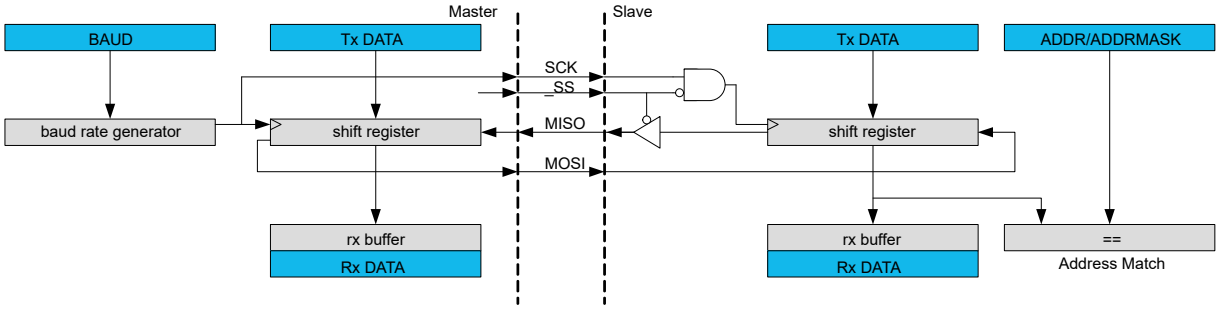
1. For t_{SCK} and t_{SSCK} values, refer to SPI Timing Characteristics.

Related Links

15.13.2 SERCOM in SPI Mode in PL0
15.13.3 SERCOM in SPI Mode in PL2
13.19 SERCOM – Serial Communication Interface
13.19.2 Features

13.21.3 Block Diagram

Figure 13-101. Full-Duplex SPI Master Slave Interconnection



13.21.4 Signal Description

Table 13-60. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links

7. I/O Multiplexing and Considerations

13.21.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.21.5.1 I/O Lines

In order to use the SERCOM's I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In master mode, the slave select line ($\overline{SS}$) is hardware controlled when the Master Slave Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 13-61. SPI Pin Configuration

Pin	Master SPI	Slave SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN=1)	Input

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the table above.

Related Links

13.17 PORT - I/O Pin Controller

13.21.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.21.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Generic Clock Controller before using the SPI.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writes to certain registers will require synchronization to the clock domains.

Related Links

13.5 GCLK - Generic Clock Controller

13.21.6.6 Synchronization

13.21.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.21.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.21.5.6 Events

Not applicable.

13.21.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.21.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

13.21.5.9 Analog Connections

Not applicable.

13.21.6 Functional Description

13.21.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

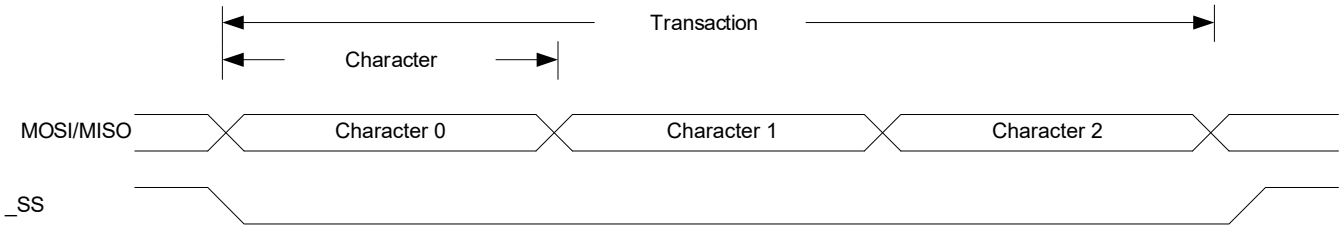
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 13-102. SPI Transaction Format



The SPI master must pull the slave select line ($\overline{SS}$) of the desired slave low to initiate a transaction. The master and slave prepare data to send via their respective shift registers, and the master generates the serial clock on the SCK line.

Data are always shifted from master to slave on the Master Output Slave Input line (MOSI); data is shifted from slave to master on the Master Input Slave Output line (MISO).

Each time character is shifted out from the master, a character will be shifted out from the slave simultaneously. To signal the end of a transaction, the master will pull the $\overline{SS}$ line high

13.21.6.2 Basic Operation

13.21.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded.

when the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in master / slave operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE= 0x2 or 0x3).
2. Select transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).

7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in master mode:
   8.1. Select the desired baud rate by writing to the Baud register (BAUD).
   8.2. If Hardware SS control is required, write '1' to the Master Slave Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

13.21.6.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

13.21.6.2.3 Clock Generation

In SPI master operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baud-rate generator.

In SPI mode, the baud-rate generator is set to synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the shift register. Refer to *Clock Generation – Baud-Rate Generator* for more details.

In SPI slave operation (CTRLA.MODE is 0x2), the clock is provided by an external master on the SCK pin. This clock is used to directly clock the SPI shift register.

Related Links

13.19.6.2.3 Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

13.21.6.2.4 Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

13.21.6.2.5 SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI data transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (Figure).

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

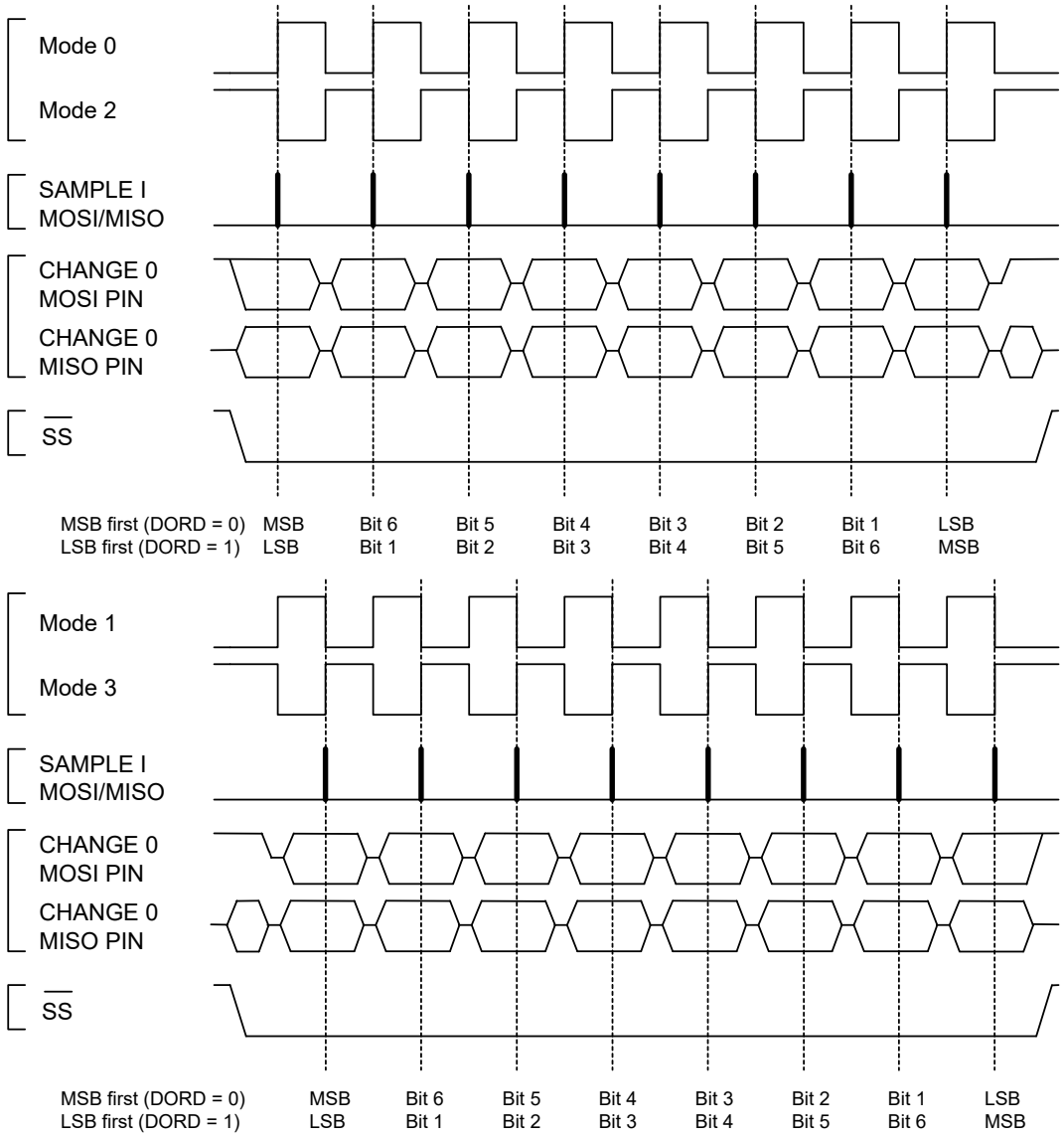
Table 13-62. SPI Transfer Modes

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Note:
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.

Figure 13-103. SPI Transfer Modes



13.21.6.2.6 Transferring Data

Master

In master mode (CTRLA.MODE=0x3), when Master Slave Enable Select (CTRLB.MSSEN) is ‘1’, hardware will control the $\overline{SS}$ line.

When Master Slave Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character will be transferred to the shift register. Once the content of TxDATA has been transferred to the shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA.

Each time one character is shifted out from the master, another character will be shifted in from the slave simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the shift register will be transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. When the transaction is finished, the master must pull the $\overline{SS}$ line high to notify the slave. If Master Slave Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

Slave

In slave mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the slave will sample and shift out data according to the transaction mode set. When the content of TxDATA has been loaded into the shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the master, the slave will receive one character for each character transmitted. A character will be transferred into the two-level receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set.

When the master pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to 13.21.6.3.2 Preloading of the Slave Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

13.21.6.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate buffer overflow notification bit in the Control A register (CTRLA.IBON):

If CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.

If CTRLA.IBON=0, the buffer overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

13.21.6.3 Additional Features

13.21.6.3.1 Address Recognition

When the SPI is configured for slave operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in sleep mode, an address match can wake up the device in order to process the transaction.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

Related Links

13.19.6.3.1 Address Match and Mask

13.21.6.3.2 Preloading of the Slave Shift Register

When starting a transaction, the slave will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

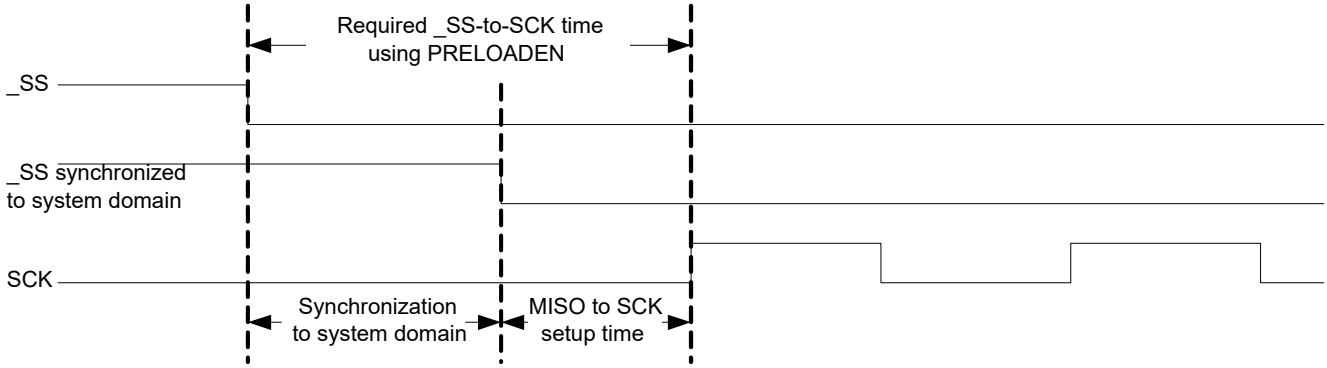
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also *Electrical Characteristics* for timing details.

Preloading is enabled by writing '1' to the Slave Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 13-104. Timing Using Preloading



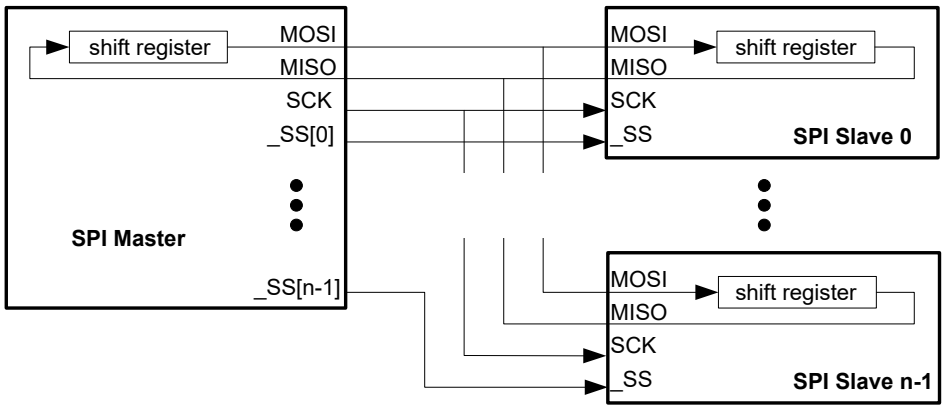
Related Links

15. Electrical Characteristics

13.21.6.3.3 Master with Several Slaves

Master with multiple slaves in parallel is only available when Master Slave Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI slaves, an SPI master can use general purpose I/O pins to control the $\overline{SS}$ line to each of the slaves on the bus, as shown in Multiple Slaves in Parallel. In this configuration, the single selected SPI slave will drive the tri-state MISO line.

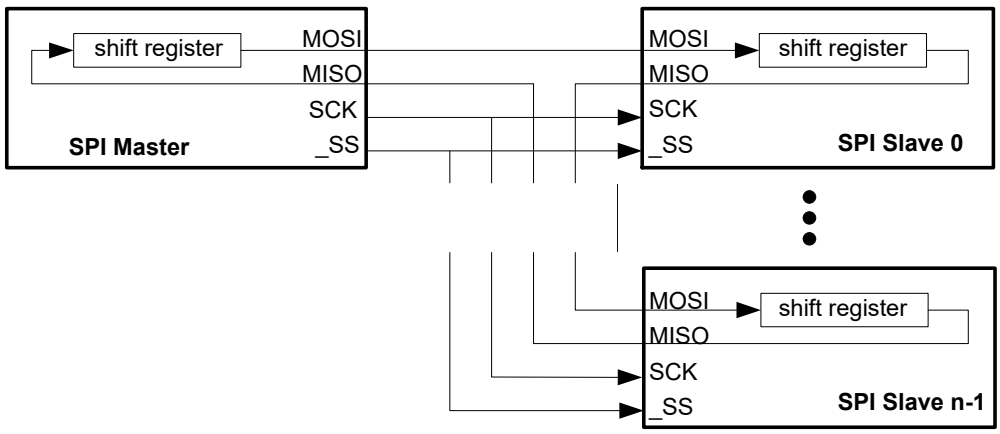
Figure 13-105. Multiple Slaves in Parallel



Another configuration is multiple slaves in series, as in Multiple Slaves in Series. In this configuration, all n attached slaves are connected in series. A common $\overline{SS}$ line is provided to all slaves, enabling them simultaneously. The

master must shift n characters for a complete transaction. Depending on the Master Slave Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 13-106. Multiple Slaves in Series



13.21.6.3.4 Loop-Back Mode

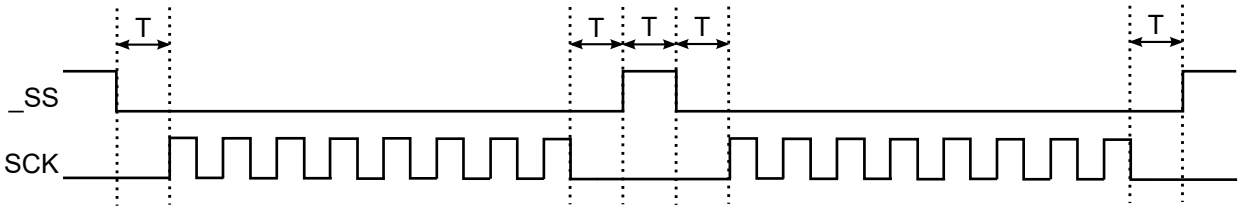
For loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

13.21.6.3.5 Hardware Controlled $\overline{SS}$

In master mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Master Slave Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI transfer mode.

Figure 13-107. Hardware Controlled $\overline{SS}$



T = 1 to 2 baud cycles

When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

13.21.6.3.6 Slave Select Low Detection

In slave mode, the SPI can wake the CPU when the slave select ($\overline{SS}$) goes low. When the Slave Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Slave Select Low interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

13.21.6.4 DMA, Interrupts, and Events

Table 13-63. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Slave Select low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

13.21.6.4.1 DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

13.21.6.4.2 Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Slave Select Low (SSL)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SPI has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.21.6.4.3 Events

Not applicable.

13.21.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_CORE will continue to run in idle sleep mode and in standby sleep mode. Any interrupt can wake up the device.
- Master operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

13.21.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also 13.21.8.2 CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

13.21.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								IBON
		23:16			DIPO[1:0]				DOPO[1:0]	
		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
		15:8	AMODE[1:0]		MSSEN				SSDE	
		23:16							RXEN	
		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
		15:8								
		23:16	ADDRMASK[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8								DATA[8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

13.21.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Refer to 13.21.6.6 Synchronization

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to 13.21.5.8 Register Access Protection.

13.21.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
								IBON
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the shift register.
This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI transfer mode.
This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI transfer mode.
This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data is sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data is sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0] Frame Format
This bit field selects the various frame formats supported by the SPI in slave mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout
These bits define the data in (DI) pad configurations.
In master operation, DI is MISO.
In slave operation, DI is MOSI.
These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout
This bit defines the available pad configurations for data out (DO) and the serial clock (SCK). In slave operation, the slave select line ($\overline{SS}$) is controlled by DOPO, while in master operation the $\overline{SS}$ line is controlled by the port configuration.
In master operation, DO is MOSI.
In slave operation, DO is MISO.
These bits are not synchronized.

DOPO	DO	SCK	Slave $\overline{SS}$	Master $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Bit 8 – IBON Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.
This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby
This bit defines the functionality in standby sleep mode.
These bits are not synchronized.

RUNSTDBY	Slave	Master
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.
0x2: SPI slave operation
0x3: SPI master operation

These bits are not synchronized.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled.
The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing "1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.21.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	
Access							R/W	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.
Writing '1' to CTRLB.RXEN when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled CTRLB.RXEN will read back as '1'.
Writing '1' to CTRLB.RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the slave addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in master mode.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	-	Reserved

Bit 13 – MSSEN Master Slave Select Enable
This bit enables hardware slave select ($\overline{SS}$) control.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE Slave Select Low Detect Enable
This bit enables wake up when the slave select ($\overline{SS}$) pin transitions from high to low.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN Slave Data Preload Enable
Setting this bit will enable preloading of the slave shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the shift register.

Bits 2:0 – CHSIZE[2:0] Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

13.21.8.3 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BAUD[7:0] Baud Register
These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator*.

Related Links

13.19.6.2.3 Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

13.21.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Slave Select Low Interrupt Enable bit, which disables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

13.21.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Slave Select Low Interrupt Enable bit, which enables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

13.21.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The BUFOVF error will set this interrupt flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 3 – SSL Slave Select Low
This flag is cleared by writing '1' to it.
This bit is set when a high to low transition is detected on the _SS pin in slave mode and Slave Select Low Detect (CTRLB.SSDE) is enabled.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.
This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction will be an address.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
In master mode, this flag is set when the data have been shifted out and there are no new data in DATA.
In slave mode, this flag is set when the _SS pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready for new data to transmit.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

13.21.8.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BUFOVF		
Access						R/W		
Reset						0		

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading DATA will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when a buffer overflow condition is detected. See also CTRLA.IBON for overflow handling.
When set, the corresponding RxDATA will be zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

13.21.8.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB CTRLB Synchronization Busy
Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB=1, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete.
Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete.
Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

13.21.8.9 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ADDRMASK[7:0] Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

13.21.8.10 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
								DATA[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0] Data
Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.
Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

13.21.8.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

13.22 SERCOM I²C – SERCOM Inter-Integrated Circuit

13.22.1 Overview

The inter-integrated circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 13-108. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM. Each master and slave have a separate I²C interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C master uses the SERCOM baud-rate generator, while the I²C slave uses the SERCOM address match logic.

Related Links
13.19 SERCOM – Serial Communication Interface

13.22.2 Features

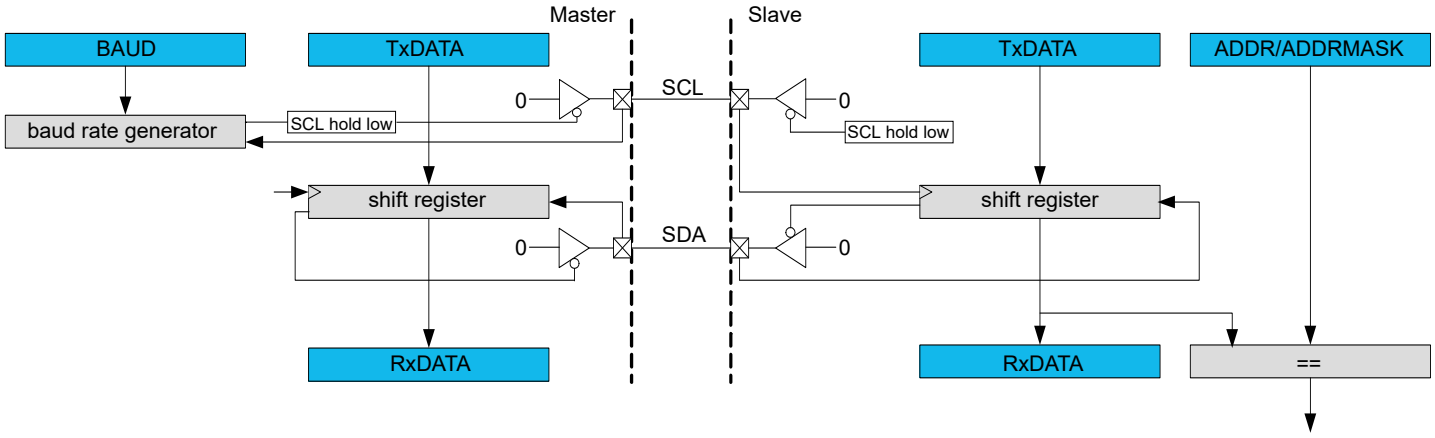
SERCOM I²C includes the following features:

- Master or slave operation
- Can be used with DMA
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode low system clock frequencies
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Slave operation:
  - Operation in all sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
  - • Unique address and/or 7-bit general call address
    • Address range
    • Two unique addresses can be used with DMA

Related Links
13.19.2 Features

13.22.3 Block Diagram

Figure 13-108. I²C Single-Master Single-Slave Interconnection



13.22.4 Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (4-wire)
PAD[3]	Digital I/O	SDC_OUT (4-wire)

One signal can be mapped on several pins.

Not all the pins are I²C pins.

Related Links

7. I/O Multiplexing and Considerations

13.22.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.22.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Related Links

13.17 PORT - I/O Pin Controller

13.22.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.22.5.3 Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock Controller and the Power Manager.

Two generic clocks ared used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a master. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I^2C.

These generic clocks are asynchronous to the bus clock (CLK_SERCOMx_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to 13.22.6.6 Synchronization for further details.

Related Links

13.5 GCLK - Generic Clock Controller

13.8 PM – Power Manager

13.22.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.22.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.22.5.6 Events

Not applicable.

13.22.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to the 13.22.10.11 DBGCTRL register for details.

13.22.5.8 Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

13.22.5.9 Analog Connections

Not applicable.

13.22.6 Functional Description

13.22.6.1 Principle of Operation

The I^2C interface uses two physical lines for communication:

- Serial Data Line (SDA) for packet transfer

- Serial Clock Line (SCL) for the bus clock

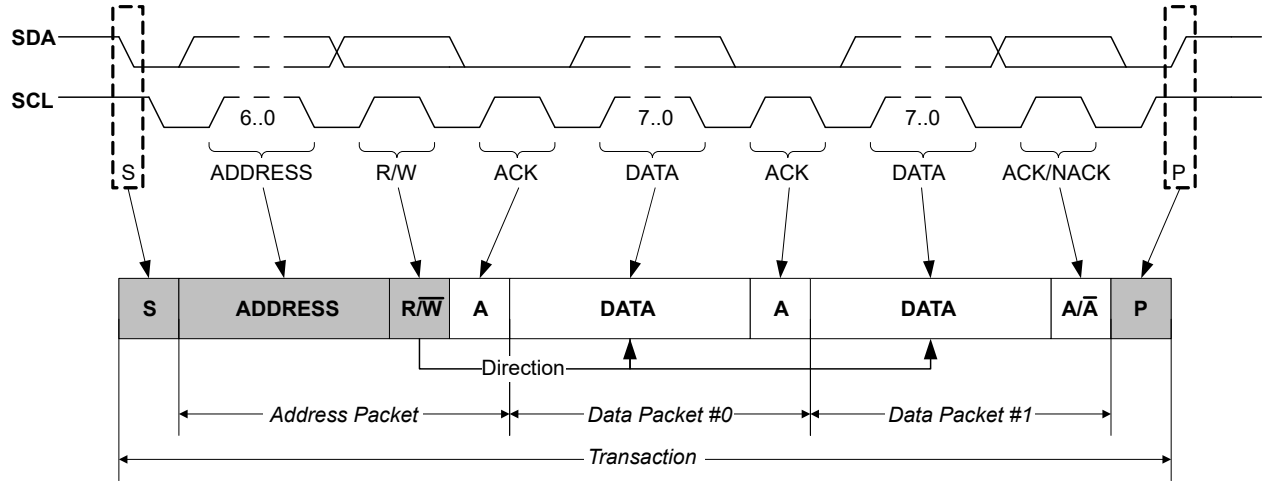
A transaction starts with the I²C master sending the start condition, followed by a 7-bit address and a direction bit (read or write to/from the slave).

The addressed I²C slave will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is not acknowledged (NACK), whether by the I²C slave or master, the I²C master takes action by either terminating the transaction by sending the stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

Figure 13-109. Basic I²C Transaction Diagram



Transaction Diagram Symbols

Bus Driver

Master driving bus

Slave driving bus

Either Master or Slave driving bus

Special Bus Conditions

S START condition

Sr repeated START condition

P STOP condition

Data Package Direction

R Master Read '1'

W Master Write '0'

Acknowledge

A Acknowledge (ACK) '0'

Ā Not Acknowledge (NACK) '1'

13.22.6.2 Basic Operation

13.22.6.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is '0'):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in slave operation.

When the I^2C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I^2C is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the I^2C is enabled it must be configured as outlined by the following steps:

1. Select I^2C Master or Slave mode by writing 0x4 or 0x5 to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
3. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
4. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
5. In Master mode:
   5.1. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   5.2. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Slave mode:
   5.1. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).
   5.2. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

13.22.6.2.2 Enabling, Disabling, and Resetting

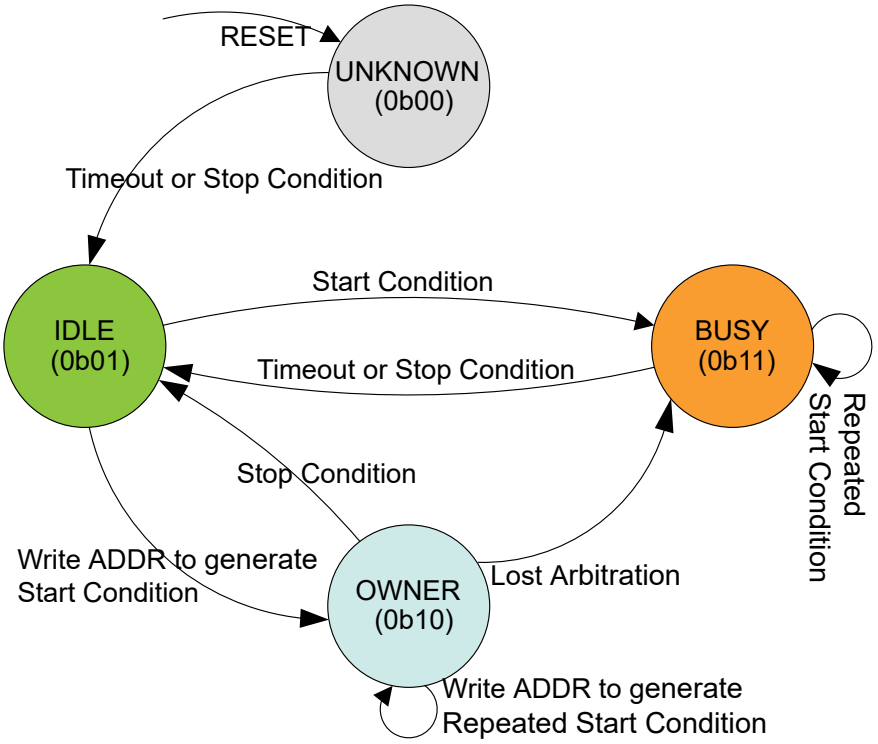
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Refer to 13.22.10.1 CTRLA for details.

13.22.6.2.3 I^2C Bus State Logic

The bus state logic includes several logic blocks that continuously monitor the activity on the I^2C bus lines in all sleep modes. The start and stop detectors and the bit counter are all essential in the process of determining the current bus state. The bus state is determined according to Bus State Diagram. Software can get the current bus state by reading the Master Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

Figure 13-110. Bus State Diagram



The bus state machine is active when the I²C master is enabled.

After the I²C master has been enabled, the bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by by writing 0b01 to STATUS.BUSSTATE
- A stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

Note: Once a known bus state is established, the bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE it is ready for a new transaction. If a start condition is issued on the bus by another I²C master in a multi-master setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either when a stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I²C master can issue a stop condition, which will change the bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the bus state becomes BUSY until a stop condition is detected. A repeated start condition will change the bus state only if arbitration is lost while issuing a repeated start.

Regardless of winning or losing arbitration, the entire address will be sent. If arbitration is lost, only 'ones' are transmitted from the point of losing arbitration and the rest of the address length.

Note: Violating the protocol may cause the I²C to hang. If this happens it is possible to recover from this state by a software reset (CTRLA.SWRST='1').

Related Links

13.22.10.1 CTRLA

13.22.6.2.4 I²C Master Operation

The I²C master is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of

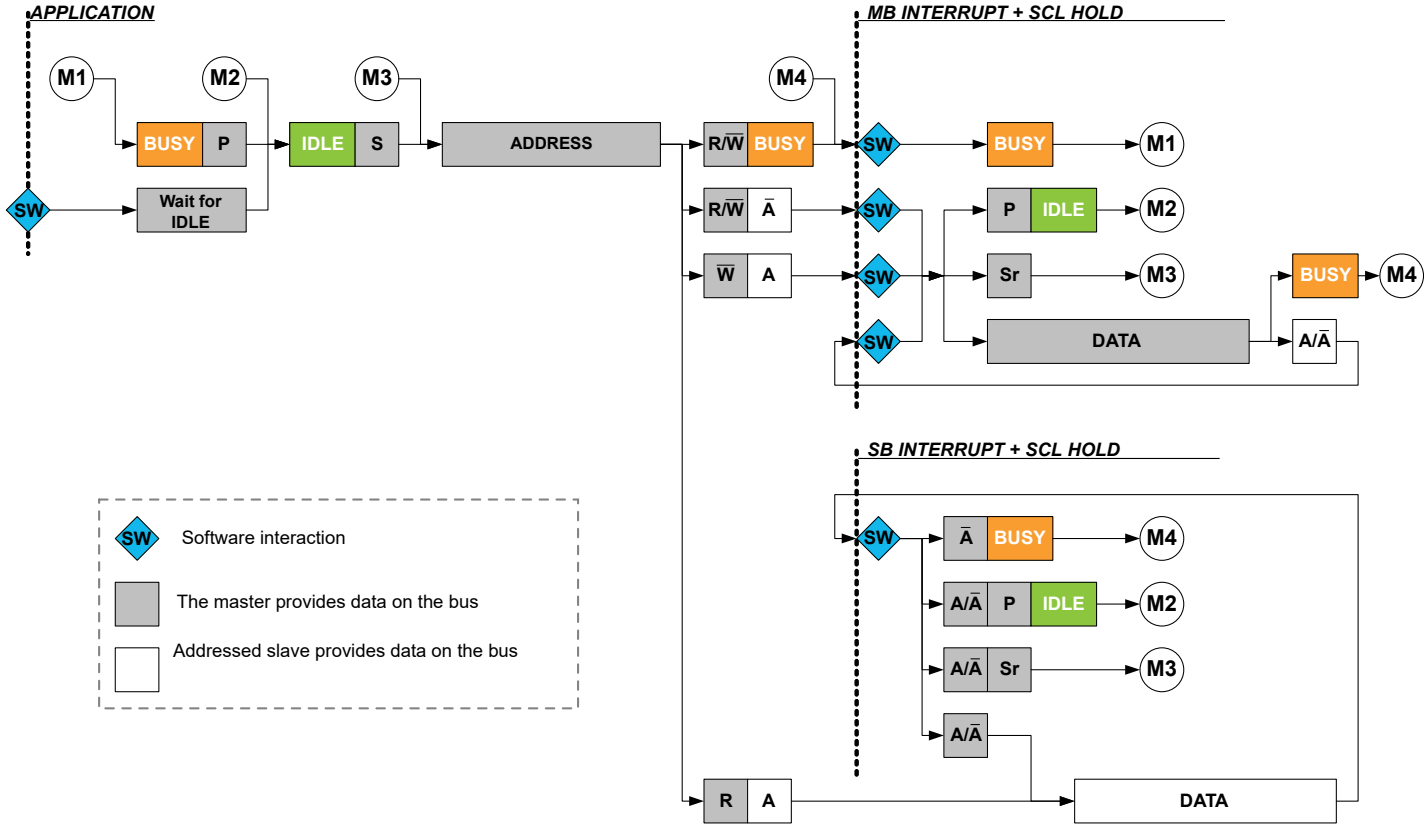
operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C master has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit . In this mode the I²C master operates according to Master Behavioral Diagram (SCLSM=0). The circles labelled "M*n*" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C master operation throughout the document.

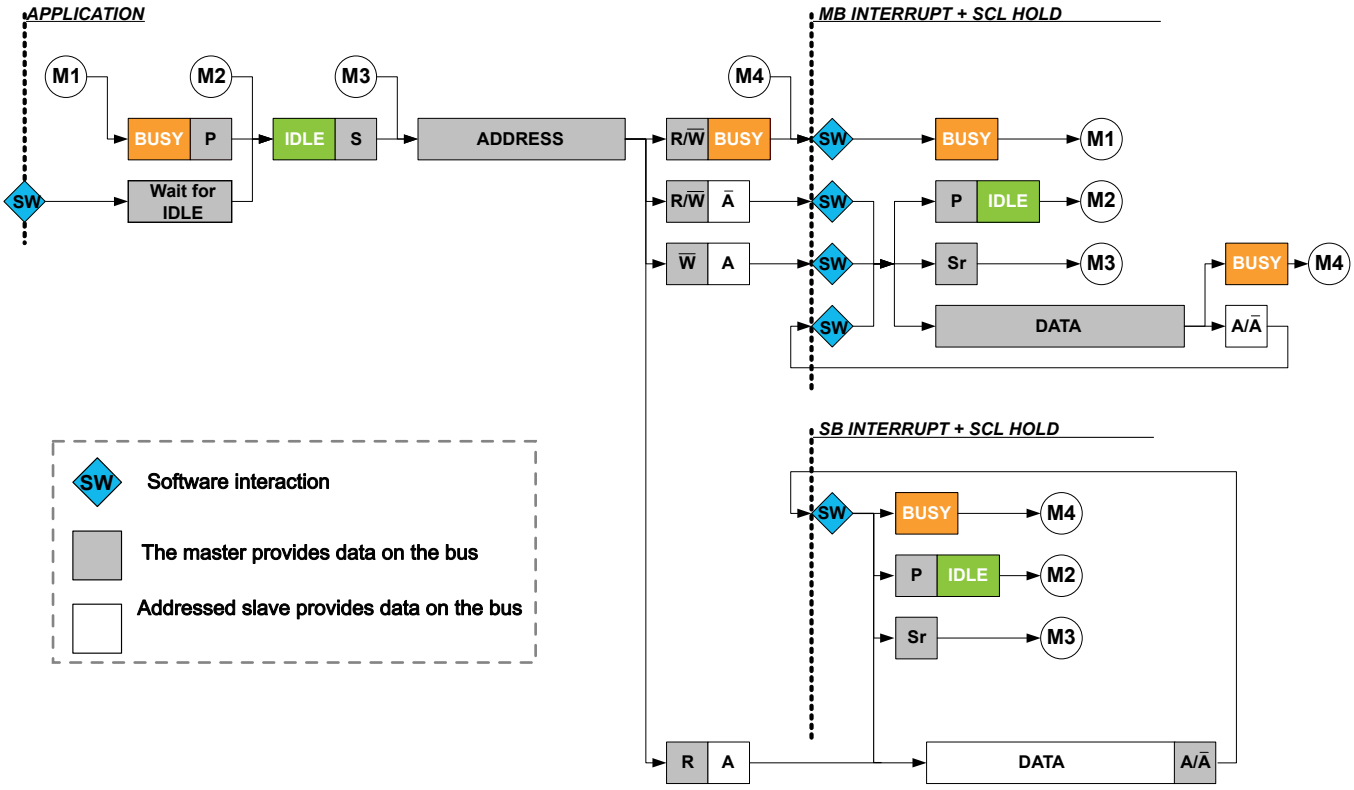
Figure 13-111. I²C Master Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in Master Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Figure 13-112. I²C Master Behavioral Diagram (SCLSM=1)



Master Clock Generation

The SERCOM peripheral supports several I²C bi-directional modes:

- Standard mode (*Sm*) up to 100kHz
- Fast mode (*Fm*) up to 400kHz
- Fast mode Plus (*Fm+*) up to 1MHz
- High-speed mode (*Hs*) up to 3.4MHz

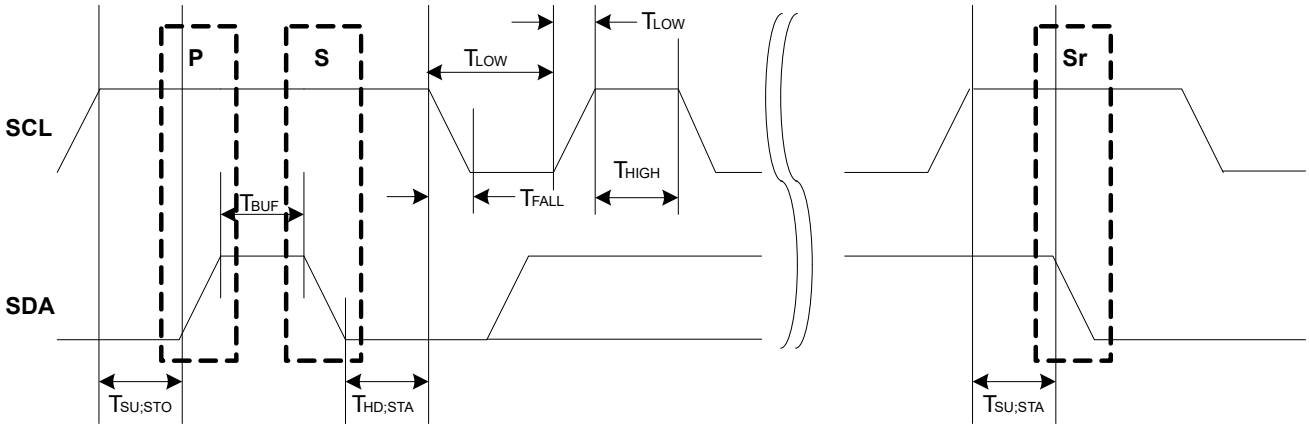
The Master clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus). For *Hs*, refer to Master Clock Generation (High-Speed Mode).

Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, the Master clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} will be considered as part of T_{LOW}. Likewise, T_{RISE} will be in a state between T_{LOW} and T_{HIGH} until a high state has been detected.

Figure 13-113. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Master Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Master Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups. Refer to *Electrical Characteristics*.
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero. Refer to *Electrical Characteristics* for details.

The SCL frequency is given by:

$$f_{\mathrm{SCL}} = \frac{1}{T_{\mathrm{LOW}} + T_{\mathrm{HIGH}} + T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + 2BAUD + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + BAUD + BAUDLOW + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{\mathrm{LOW}} = \frac{BAUDLOW + 5}{f_{\mathrm{GCLK}}}$$

$$T_{\mathrm{HIGH}} = \frac{BAUD + 5}{f_{\mathrm{GCLK}}}$$

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Related Links

15. Electrical Characteristics

Master Clock Generation (High-Speed Mode)
For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

Transmitting Address Packets

The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in 13.22.6.1 Principle of Operation. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C slave device responding to the address packet, then the INTFLAG.MB interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C slave is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a stop condition (recommended) or resending the address packet by a repeated start condition. When using SMBus logic, the slave must ACK the address. If there is no response, it means that the slave is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Master on Bus set

If the I²C master receives an acknowledge response from the I²C slave, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated start condition will automatically be inserted before the address packet.
- Issue a stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Slave on Bus set

If the I2C master receives an ACK from the I2C slave, the I2C master proceeds to receive the next byte of data from the I2C slave. When the first data byte is received, the Slave on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I2C operation to continue:

- Let the I2C master continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a stop condition.

Note: An ACK or NACK will be automatically transmitted if smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

Transmitting Data Packets

When an address packet with direction Master Write (see Figure 13-109) was transmitted successfully , INTFLAG.MB will be set. The I2C master will start transmitting data via the I2C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I2C master will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I2C master will receive an ACK bit from the I2C slave, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I2C Master on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I2C master is not allowed to continue transmitting data packets if a NACK is received from the I2C slave.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I2C master will already have received one data packet. The I2C master must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I2C master will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the smart mode.

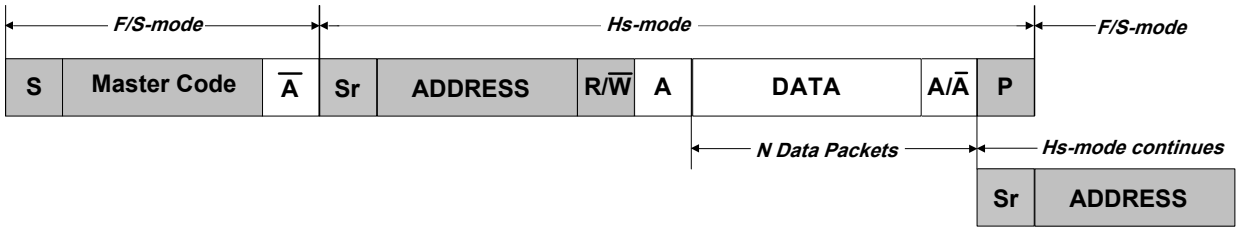
High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a master code (0b00001nnn, where 'nnn' is a unique master code) is transmitted in Full-speed mode, followed by a NACK since no slaveshould acknowledge. Arbitration is performed only during the Full-speed Master Code phase. The master code is transmitted by writing the master code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the master code and NACK have been transmitted, the master write interrupt will be asserted. In the meanwhile, the slave address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the master will generate a repeated start, followed by the slave address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 13-114. High Speed Transfer



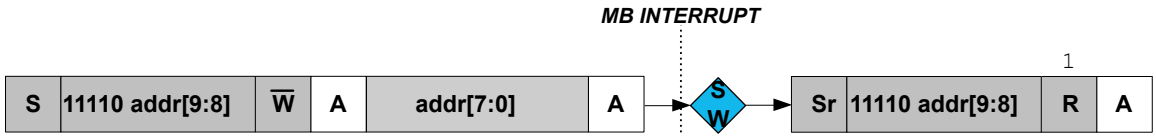
Transmitting in High-speed mode requires the I²C master to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed slave acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the master must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the master receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more slaves, then the master will proceed to transmit the second address byte and the master will first see the write interrupt flag after the second byte is transmitted. If the transaction direction is read-from-slave, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 13-115. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Master on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

13.22.6.2.5 I²C Slave Operation

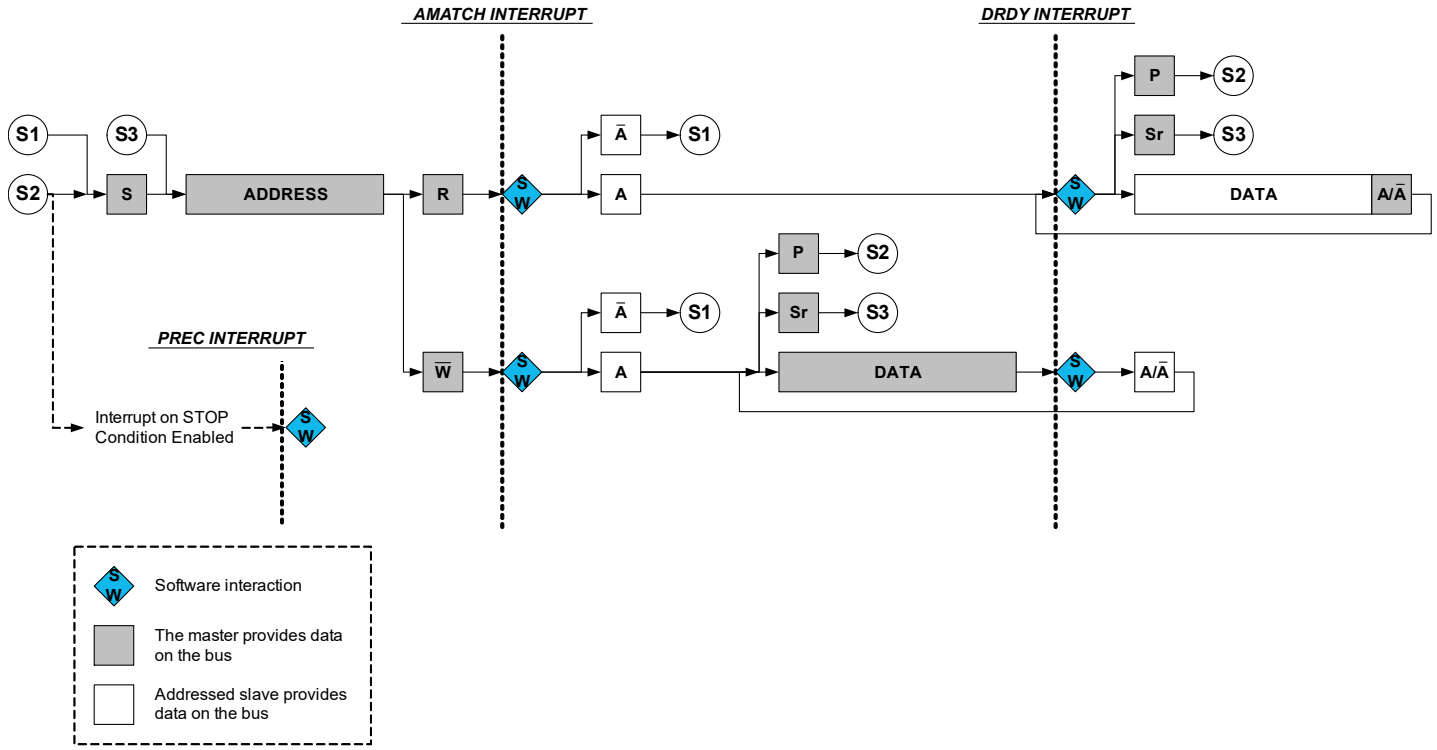
The I²C slave is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C slave has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C slave operates according to I²C Slave Behavioral Diagram (SCLSM=0). The circles labelled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C slave operation throughout the document.

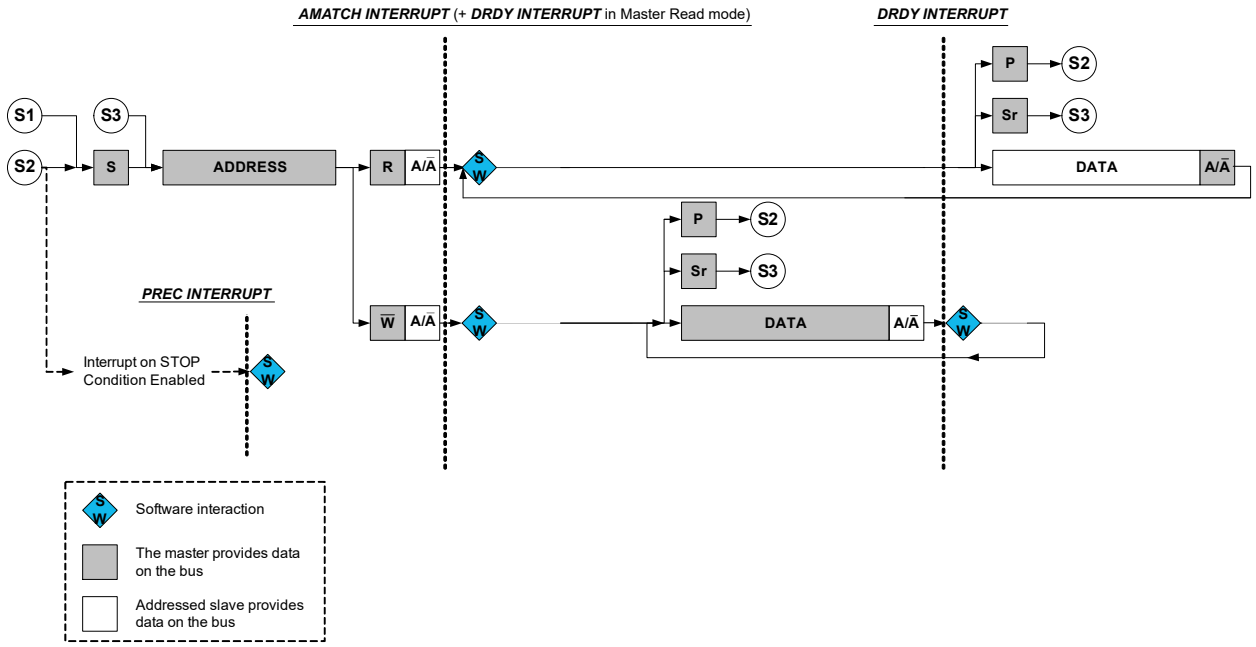
Figure 13-116. I²C Slave Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Slave Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging. For master reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for master writes, the first data interrupt will be seen after the first data byte has been received by the slave and the acknowledge bit has been sent to the master.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 13-117. I²C Slave Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 13-116. When the I²C slave is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C slave will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C master, one of two cases will arise based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C slave hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C slave will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I²C slave Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I²C master write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C slave will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C slave will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C slave command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I²C slave will stretch the SCL line only after an ACK, see Slave Behavioral Diagram (SCLSM=1). When the I²C slave is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I²C slave will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I²C master, INTFLAG.AMATCH be set to '1' to clear it.

Receiving and Transmitting Data Packets

After the I²C slave has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C slave will send an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C slave must expect a stop or a repeated start to be received. The I²C slave must release the data line to allow the I²C master to generate a stop or repeated start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C slave will return to IDLE state.

High-Speed Mode

When the I²C slave is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the slave recognizes a START followed by a master code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The slave will then remain in High-speed mode until a STOP is received.

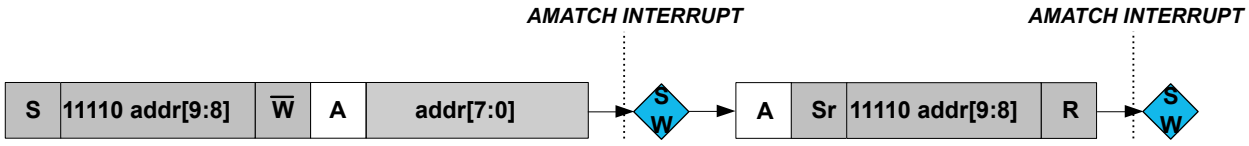
10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit slave address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The slave matches on the second address as it it was addressed by the previous 10-bit address.

Figure 13-118. 10-bit Addressing



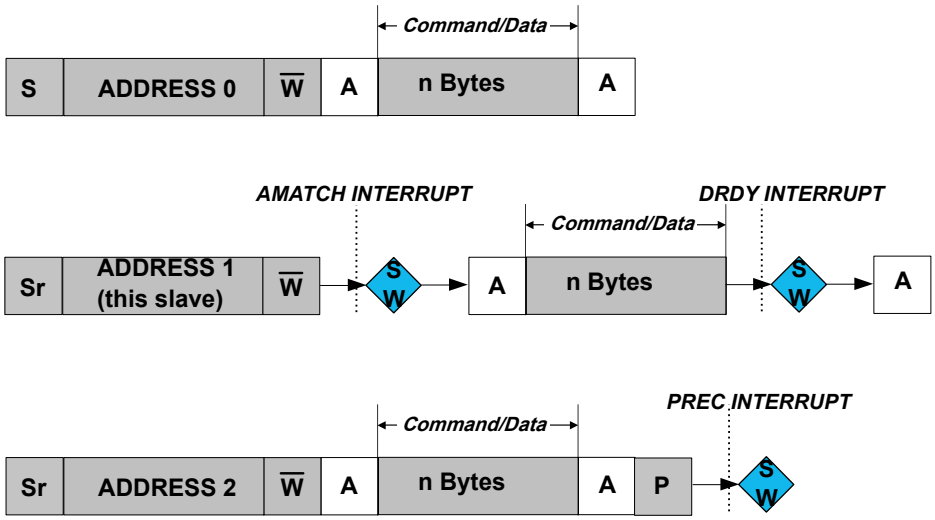
PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set when a STOP condition is detected on the bus. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the slaves addressed during the group command, they all begin executing the command they received.

PMBus Group Command Example shows an example where this slave, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple slaves addressed before and after this slave. Eventually, at the end of the group command, a single STOP is generated by the master. At this point a STOP interrupt is asserted.

Figure 13-119. PMBus Group Command Example



13.22.6.3 Additional Features

13.22.6.3.1 SMBus

The I²C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, master extend time-out, and slave extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32KHz oscillator. The I²C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a slave device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the master device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

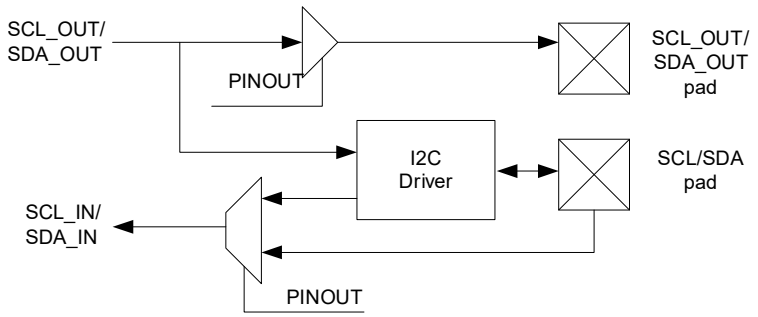
13.22.6.3.2 Smart Mode

The I²C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I²C protocol. The smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

13.22.6.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I²C tri-state drivers are bypassed, and an external I²C compliant tri-state driver is needed when connecting to an I²C bus.

Figure 13-120. I²C Pad Interface



13.22.6.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

13.22.6.4 DMA, Interrupts and Events

Table 13-64. Module Request for SERCOM I²C Slave

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Slave transmit mode)	Yes (request cleared when data is written)		NA
Data received (RX) (Slave receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
Error (ERROR)		Yes	

Table 13-65. Module Request for SERCOM I²C Master

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Master transmit mode)	Yes (request cleared when data is written)		NA
Data needed for transmit (RX) (Master transmit mode)	Yes (request cleared when data is read)		
Master on Bus (MB)		Yes	
Stop received (SB)		Yes	
Error (ERROR)		Yes	

13.22.6.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

Slave DMA

When using the I²C slave with DMA, an address match will cause the address interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C slave generates the following requests:

- Write data received (RX): The request is set when master write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX): The request is set when data is needed for a master read operation. The request is cleared when DATA is written.

Master DMA

When using the I²C master with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN), and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for master reads) and a STOP.

If a NACK is received by the slave for a master write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I²C master generates the following requests:

- Read data received (RX): The request is set when master read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX): The request is set when data is needed for a master write operation. The request is cleared when DATA is written.

13.22.6.4.2 Interrupts

The I²C slave has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I²C master has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Slave on Bus (SB)
- Master on Bus (MB)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See 13.22.10.6 INTFLAG register for details on how to clear interrupt flags.

The I²C has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.22.6.4.3 Events

Not applicable.

13.22.6.5 Sleep Mode Operation

I²C Master Operation

The generic clock (GCLK_SERCOMx_CORE) will continue to run in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in standby sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I²C Slave Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

13.22.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation.

The following registers are synchronized when written:

- Data (DATA) when in master operation

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

13.22.7 Register Summary - I2C Slave

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN		SDAHOLD[1:0]					PINOUT
		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR					DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR					DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR					DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
		15:8						LENERR	SEXTTOUT	
0x1C	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
		15:8	TENBITEN					ADDR[9:7]		
		23:16	ADDRMASK[6:0]							
		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
		15:8								

13.22.8 Register Description - I2C Slave

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.22.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.22.6.6 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.22.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the slave will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 13-116
1	SCL stretch only after ACK bit according to Figure 13-117

Bits 25:24 – SPEED[1:0] Transfer Speed
These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the slave will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

Bit 16 – PINOUT Pin Usage
This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bit 7 – RUNSTDBY Run in Standby
This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x04 to select the I^2C slave serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled.
The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.22.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action
This bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.
This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command
This bit field triggers the slave operation as the below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the slave interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.
All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.
This bit is not enable-protected.

Table 13-66. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Master read)	Wait for any start (S/Sr) condition

.....continued		
CMD[1:0]	**DIR**	**Action**
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute acknowledge action succeeded by slave data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0] Address Mode
These bits set the addressing mode.
These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The slave responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	-	Reserved.

Bit 10 – AACKEN Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.
This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the slave has been addressed since the last STOP condition on the bus.
This bit is not write-synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable
When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.
This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

Related Links
13.19 SERCOM – Serial Communication Interface

13.22.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

13.22.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

13.22.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are SEXTTOUT, LOWTOUT, COLL, and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 2 – DRDY Data Ready
This flag is set when a I^2C slave byte transmission is successfully completed.
The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready interrupt flag.

Bit 1 – AMATCH Address Match
This flag is set when the I^2C slave address match logic detects that a valid address has been received.
The flag is cleared by hardware when CTRL.CMD is written.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received
This flag is set when a stop condition is detected for a transaction being processed. A stop condition detected between a bus master and another slave will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).
This flag is cleared by hardware after a command is issued on the next address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received interrupt flag.

13.22.8.6 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	
Access						R/W	R/W	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 10 – LENERR Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Bit 10 – HS High-speed
This bit is set if the slave detects a START followed by a Master Code transmission.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

Bit 9 – SEXTTOUT Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD Clock Hold
The slave Clock Hold bit (STATUS.CLKHOLD) is set when the slave is holding the SCL line low, stretching the I2C clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.
This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT SCL Low Time-out
This bit is set if an SCL low time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR Repeated Start
When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.
This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR Read / Write Direction
The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a master.

Value	Description
0	Master write operation is in progress.
1	Master read operation is in progress.

Bit 2 – RXNACK Received Not Acknowledge
This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Master responded with ACK.
1	Master responded with NACK.

Bit 1 – COLL Transmit Collision
If set, the I2C slave was not able to transmit a high data or NACK bit, the I2C slave will immediately release the SDA and SCL lines and wait for the next packet addressed to it.
This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.
Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR Bus Error
The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.
This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.
Writing a '1' to this bit will clear the status.
Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

13.22.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.
Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.
Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

13.22.8.8 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
	TENBITEN					ADDR[9:7]		
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:17 – ADDRMASK[9:0] Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bit 15 – TENBITEN Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

Bits 10:1 – ADDR[9:0] Address
These bits contain the I^2C slave address used by the slave address match logic to determine if a master has addressed the slave.
When using 7-bit addressing, the slave address is represented by ADDR[6:0].
When using 10-bit addressing (ADDR.TENBITEN=1), the slave address is represented by ADDR[9:0]
When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (master write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

13.22.8.9 Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0] Data
The slave data register I/O location (DATA.DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the slave (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.
Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).
Writing or reading DATA.DATA when not in smart mode does not require synchronization.

13.22.9 Register Summary - I2C Master

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8							QCEN	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		23:16	HSBAUD[7:0]							
		31:24	HSBAUDLOW[7:0]							
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR						SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR						SB	MB
0x18	DATA	7:0	DATA[7:0]							
		15:8								
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0								
		15:8	TENBITEN	HS	LENEN			ADDR[2:0]		
		23:16	LEN[7:0]							
		31:24								
0x28 ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

13.22.10 Register Description - I2C Master

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.22.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.22.6.6 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.22.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the master will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted. INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0] Inactive Time-Out
If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an I^2C master or slave is holding the SCL low.
Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.
Calculated time-out periods are based on a 100kHz baud rate.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 13-111.
1	SCL stretch only after ACK bit, Figure 13-112.

Bits 25:24 – SPEED[1:0] Transfer Speed
These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.
SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN Master SCL Low Extend Time-Out
This bit enables the master SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.
SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT Pin Usage
This bit set the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bit 7 – RUNSTDBY Run in Standby
This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C master will not operate in standby sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all sleep modes.

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x5 to select the I^2C master serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled.
The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.22.10.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action
This bit defines the I^2C master's acknowledge behavior after a data byte is received from the I^2C slave. The acknowledge action is executed when a command is written to CTRLB.CMD, or if smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.
This bit is not enable-protected.
This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command
Writing these bits triggers a master operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in master read mode. In master write mode, a command will only result in a repeated start or stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.
Commands can only be issued when either the Slave on Bus interrupt flag (INTFLAG.SB) or Master on Bus interrupt flag (INTFLAG.MB) is '1'.
If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.
Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 13-67. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable
When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.
This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

13.22.10.3 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HSBAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – HSBAUDLOW[7:0] High Speed Master Baud Rate Low
HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to
$\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$
HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0] High Speed Master Baud Rate
This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.
$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$

Bits 15:8 – BAUDLOW[7:0] Master Baud Rate Low
If this bit field is non-zero, the SCL low time will be described by the value written.
For more information on how to calculate the frequency, see SERCOM 13.19.6.2.3 Clock Generation – Baud-Rate Generator.

Bits 7:0 – BAUD[7:0] Master Baud Rate
This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.
For more information on how to calculate the frequency, see SERCOM 13.19.6.2.3 Clock Generation – Baud-Rate Generator.

13.22.10.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Slave on Bus Interrupt Enable bit, which disables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Master on Bus Interrupt Enable bit, which disables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

13.22.10.5 Interrupt Enable Clear

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Slave on Bus Interrupt Enable bit, which enables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Master on Bus Interrupt Enable bit, which enables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

13.22.10.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 1 – SB Slave on Bus
The Slave on Bus flag (SB) is set when a byte is successfully received in master read mode, i.e., no arbitration lost or bus error occurred during the operation. When this flag is set, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and SB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

Bit 0 – MB Master on Bus
This flag is set when a byte is transmitted in master write mode. The flag is set regardless of the occurrence of a bus error or an arbitration lost condition. MB is also set when arbitration is lost during sending of NACK in master read mode, or when issuing a start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

13.22.10.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	R/W	R	R		R	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR Transaction Length Error
This bit is set when automatic length is used for a DMA transaction and the slave sends a NACK before ADDR.LEN bytes have been written by the master.
Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 9 – SEXTTOUT Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.
This bit is automatically cleared when writing to the ADDR register.
Writing '1' to this bit location will clear SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 8 – MEXTTOUT Master SCL Low Extend Time-Out
This bit is set if a master SCL low time-out occurs.
Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 7 – CLKHOLD Clock Hold
This bit is set when the master is holding the SCL line low, stretching the I^2C clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.
This bit is cleared when the corresponding interrupt flag is cleared and the next operation is given.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Bit 6 – LOWTOUT SCL Low Time-Out
This bit is set if an SCL low time-out occurs.
Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0] Bus State
These bits indicate the current I^2C bus state.

When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.
Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The bus state is unknown to the I2C master and will wait for a stop condition to be detected or wait to be forced into an idle state by software
0x1	IDLE	The bus state is waiting for a transaction to be initialized
0x2	OWNER	The I2C master is the current owner of the bus
0x3	BUSY	Some other I2C master owns the bus

Bit 2 – RXNACK Received Not Acknowledge
This bit indicates whether the last address or data packet sent was acknowledged or not.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	Slave responded with ACK.
1	Slave responded with NACK.

Bit 1 – ARBLOST Arbitration Lost
This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a start or repeated start condition on the bus. The Master on Bus interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

Bit 0 – BUSERR Bus Error
This bit indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.
If the I2C master is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.
Writing the ADDR.ADDR register will automatically clear the BUSERR flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

13.22.10.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.
Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.
Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

13.22.10.9 Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TENBITEN	HS	LENEN			ADDR[2:0]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:16 – LEN[7:0] Transaction Length
These bits define the transaction length of a DMA transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN Transfer Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:8 – ADDR[2:0] Address
When ADDR is written, the consecutive operation will depend on the bus state:
UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.
BUSY: The I^2C master will await further operation until the bus becomes IDLE.
IDLE: The I^2C master will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.

OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.
STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.
The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.
The I^2C master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

13.22.10.10 Data

Name: DATA
Offset: 0x18
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0] Data
The master data register I/O location (DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the master (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent. Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).
Writing or reading DATA.DATA when not in smart mode does not require synchronization.

13.22.10.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

13.23 TC – Timer/Counter

13.23.1 Overview

There are up to three TC peripheral instances. Up to two TCs (TC0, TC1) are in PD1, whereas TC4, present in all device configurations, is always located in power domain PD0.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events or IO pin edges, allowing for capturing of frequency and/or pulse width.

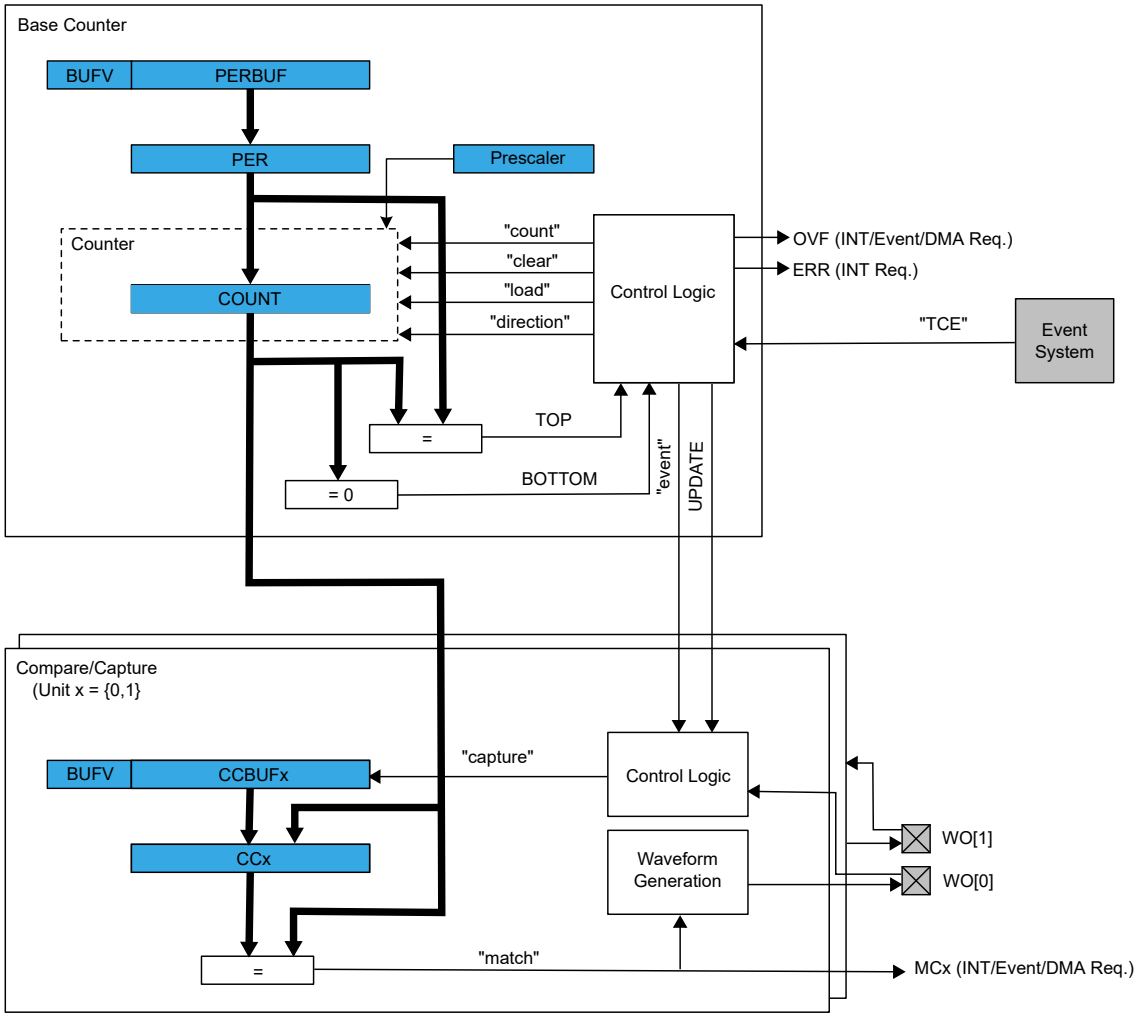
A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

13.23.2 Features

- Selectable configuration
  - 8-, 16- or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input capture
  - Event / IO pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
  - Minimum and maximum capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- DMA support

13.23.3 Block Diagram

Figure 13-121. Timer/Counter Block Diagram



13.23.4 Signal Description

Table 13-68. Signal Description for TC.

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.23.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.23.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

13.17 PORT - I/O Pin Controller

13.23.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.23.5.3 Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Main Clock Module. The default state of CLK_TCx_APB can be found in the *Peripheral Clock Masking*.

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Note that TC0 and TC1 share a peripheral clock channel, as do TC2 and TC3. For this reason they cannot be set to different clock frequencies.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.23.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.23.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.23.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

13.18 EVSYS – Event System

13.23.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

13.23.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.23.5.9 Analog Connections

Not applicable.

13.23.6 Functional Description

13.23.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 13-69. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations.
ZERO	The counter is ZERO when it contains all zeroes
MAX	The counter reaches MAX when it contains all ones
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source
Counter	The clock control is handled externally (e.g. counting external events)
CC	For compare operations, the CC are referred to as “compare channels” For capture operations, the CC are referred to as “capture channels.”

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

The Counter register (COUNT), compare and capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings determine the maximum range of the counter. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

In 8-bit mode, Period Value (PER) and Period Buffer Value (PERBUF) registers are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

13.23.6.2 Basic Operation

13.23.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE = 0):

- Control A register (13.24.8.1 CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   – If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).
5. If desired, select a one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction 'down' (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable the inversion of the waveform output or IO pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

13.23.6.2.2 Enabling, Disabling, and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disabled by writing a ‘0’ to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. Refer to the 13.23.7.1.1 CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

13.23.6.2.3 Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

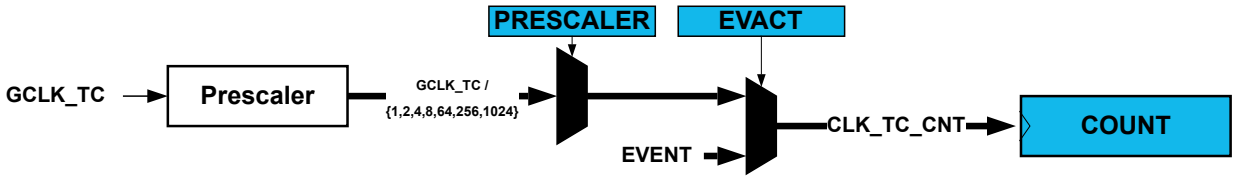
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TC_CNT.

Figure 13-122. Prescaler



13.23.6.2.4 Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8: The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16: 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32: This mode is achieved by pairing two 16-bit TC peripherals. TC0 is paired with TC1, and TC2 is paired with TC3. TC4 does not support 32-bit resolution.
  When paired, the TC peripherals are configured using the registers of the even-numbered TC (TC0 or TC2 respectively). The odd-numbered partner (TC1 or TC3 respectively) will act as slave, and the Slave bit in the Status register (STATUS.SLAVE) will be set. The register values of a slave will not reflect the registers of the 32-bit counter. Writing to any of the slave registers will not affect the 32-bit counter. Normal access to the slave COUNT and CCx registers is not allowed.

13.23.6.2.5 Counter Operations

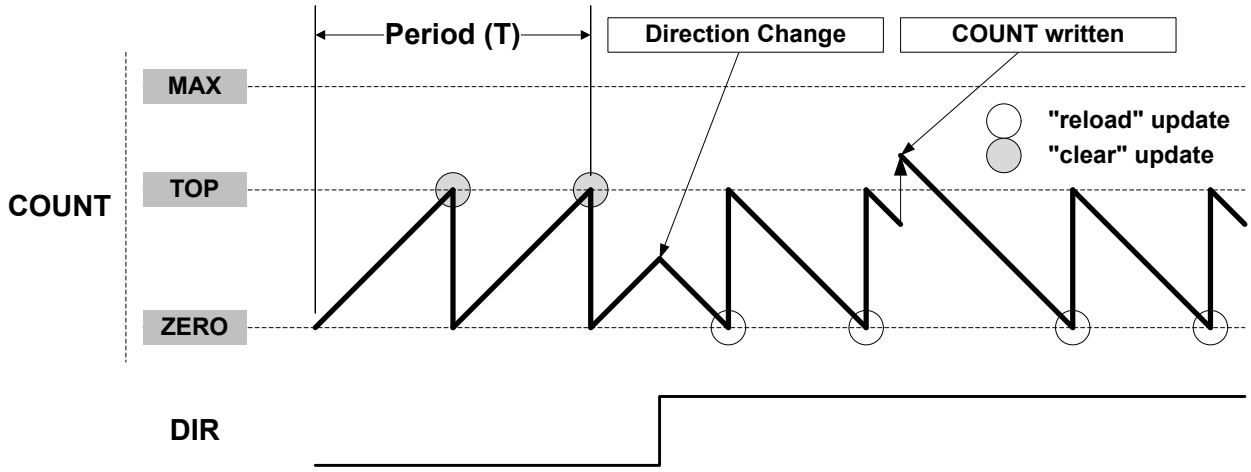
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TC_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also the figure below.

Figure 13-123. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will be loaded with the starting value (ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR). All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Count Event Action

The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Start Event Action

The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

13.23.6.2.6 Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). For further details, refer to 13.23.6.2.7 Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt/and or event can be generated on comparison match if enabled. The same condition generates a DMA request.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

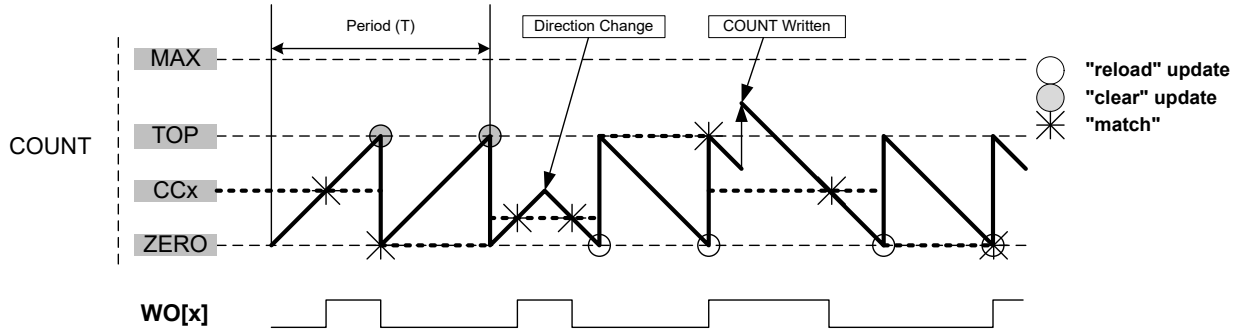
- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, the TOP will be determined by the counter resolution. In 8-bit counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In 16- and 32-bit counter mode, TOP is fixed to the maximum (MAX) value of the counter.

Normal Frequency Generation (NFRQ)

For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for 8-bit counter mode and MAX for 16- and 32-bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) will be set.

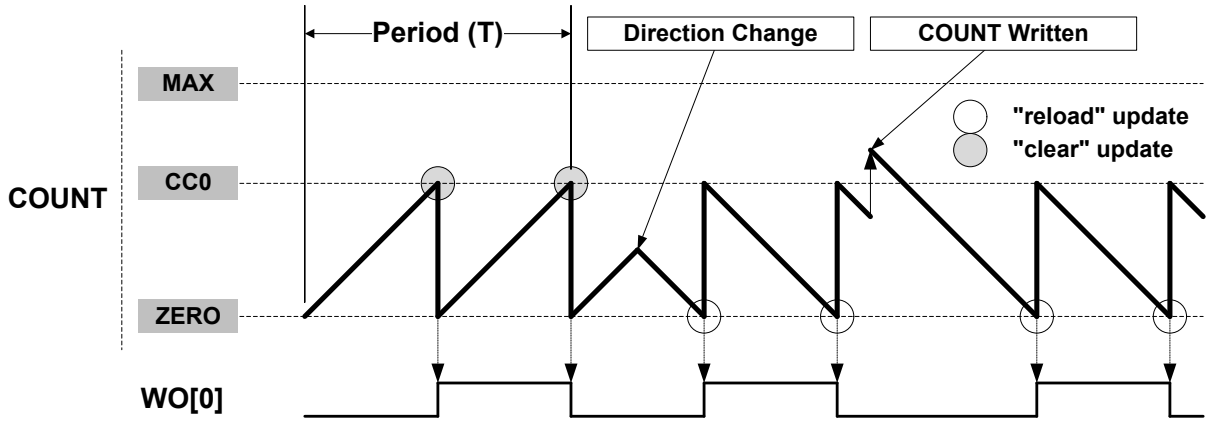
Figure 13-124. Normal Frequency Operation



Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each update condition.

Figure 13-125. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)
NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

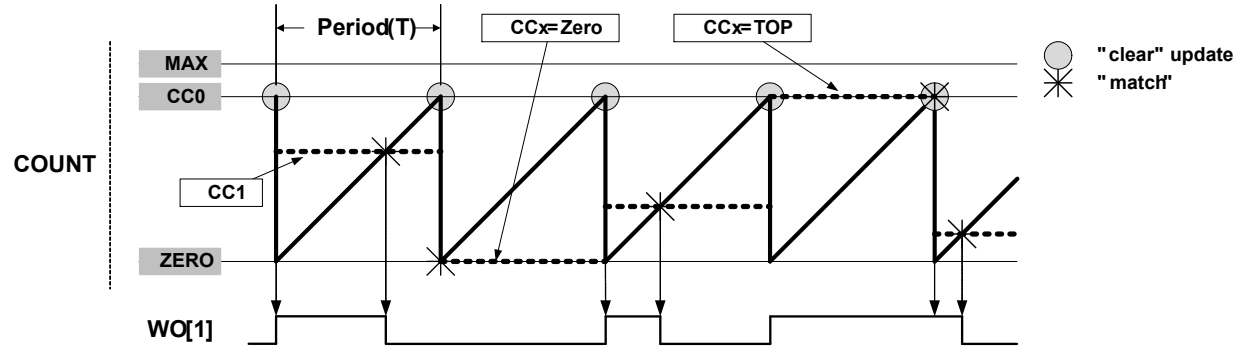
The PWM frequency (f_{PWM_SS}) depends on TOP value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TC}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)
In MPWM, the output of WO[1] is depending on CC1 as shown in the figure below. On on every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 13-126. Match PWM Operation



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 13-70. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

Related Links

13.17 PORT - I/O Pin Controller

13.23.6.2.7 Double Buffering

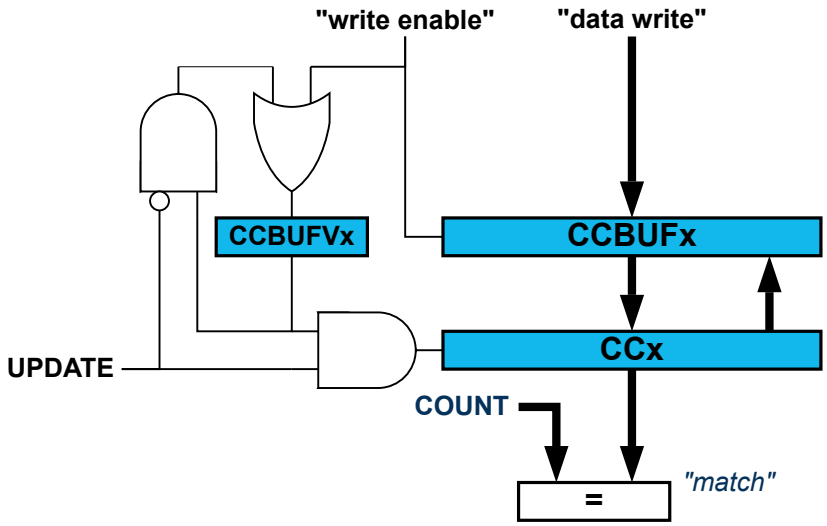
The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.

Note: The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

Figure 13-127. Compare Channel Double Buffering



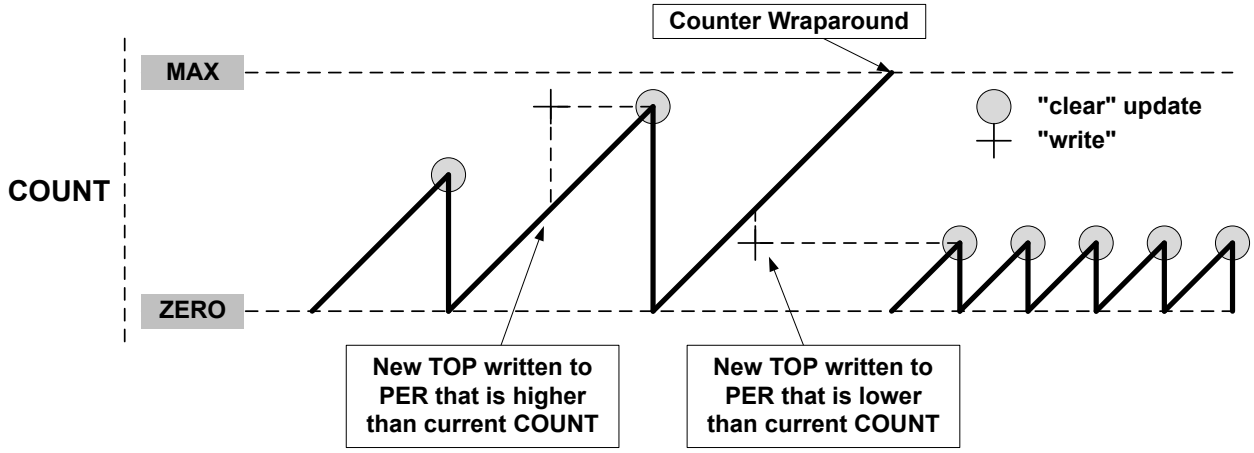
Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

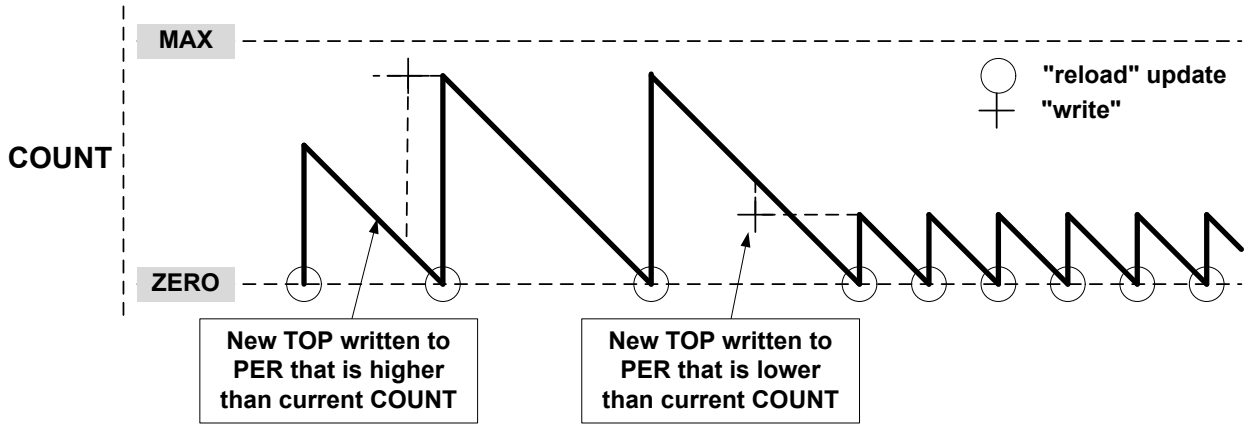
Figure 13-128. Unbuffered Single-Slope Up-Counting Operation



A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 13-128.

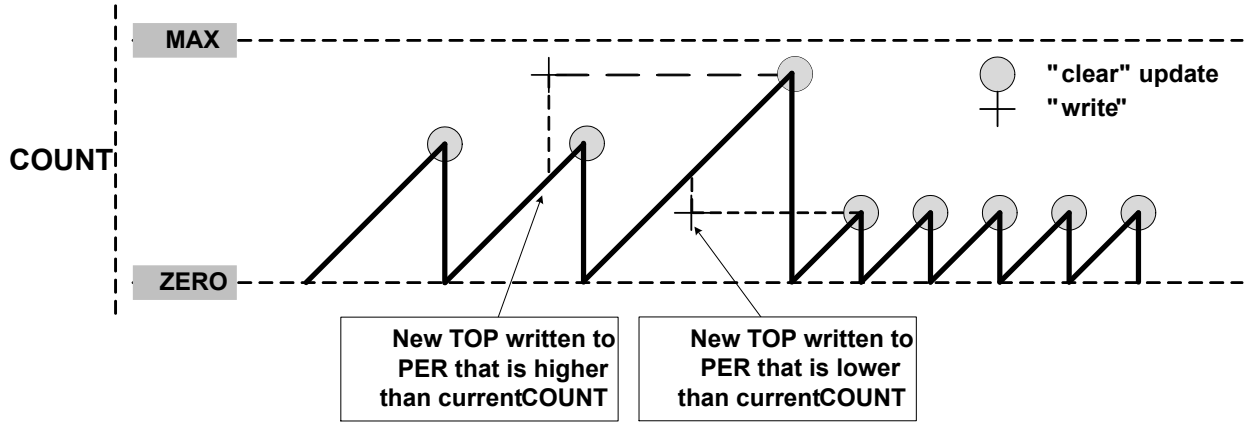
COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 13-129. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 13-130. This prevents wraparound and the generation of odd waveforms.

Figure 13-130. Changing the Period Using Buffering



13.23.6.2.8 Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by input event line TC_EV or by asynchronous IO pin WO[x] for each capture channel or by a TC event. To enable the capture from input event line, Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be written to '1'. To enable the capture from the IO pin, the Capture On Pin x Enable bit in CTRLA register (CTRLA.COPENx) must be written to '1'.

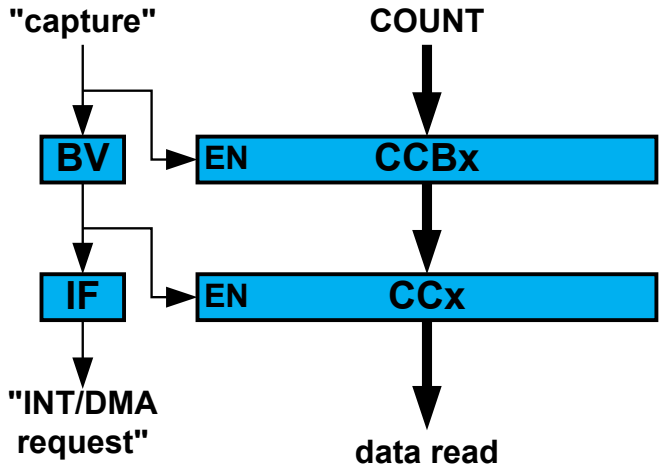
Note: The RETRIGGER, COUNT and START event actions are available only on an event from the Event System.

By default, a capture operation is done when a rising edge is detected on the input signal. Capture on falling edge is available, its activation is depending on the input source:

- When the channel is used with a IO pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Figure 13-131. Capture Double Buffering

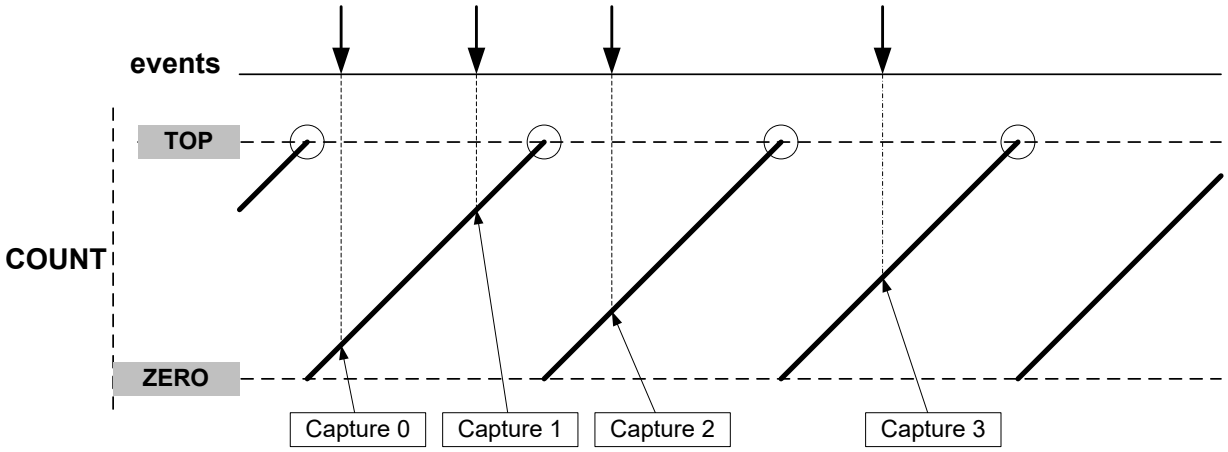


For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System or from the corresponding IO pin, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 13-132. Input Capture Timing



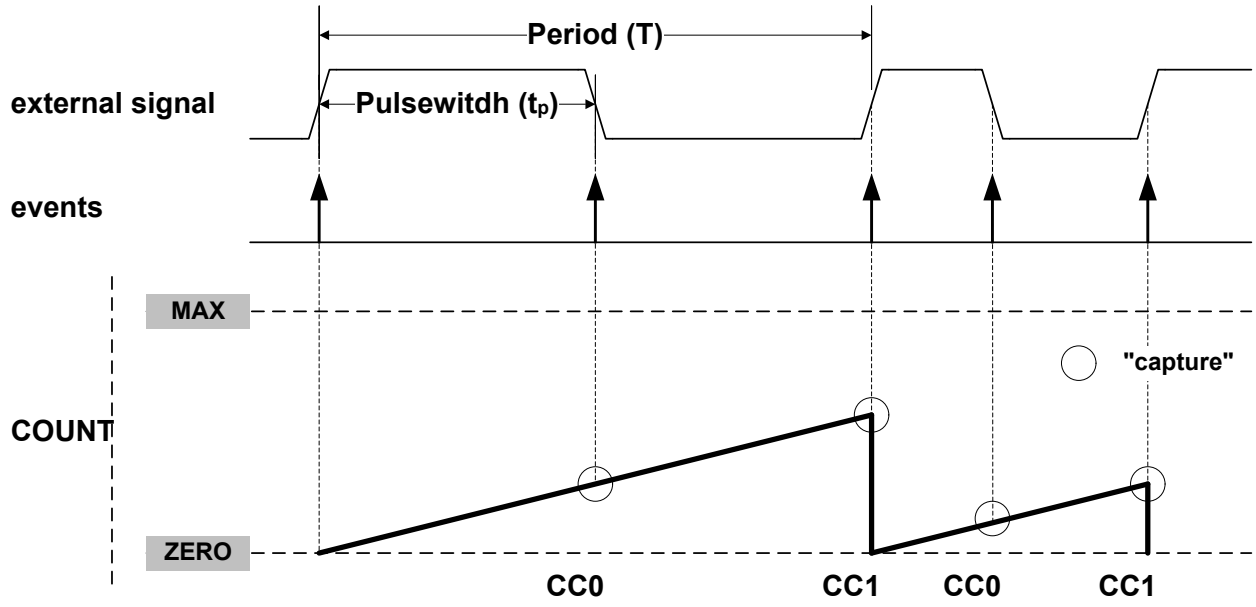
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T} \qquad \text{dutyCycle} = \frac{t_p}{T}$$

Figure 13-133. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period *T* will be captured

into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT=PPW (period and pulse-width)offers identical functionality, but will capture T into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge. This also be for DRVCTRL.INVENx if pin capture is enabled.

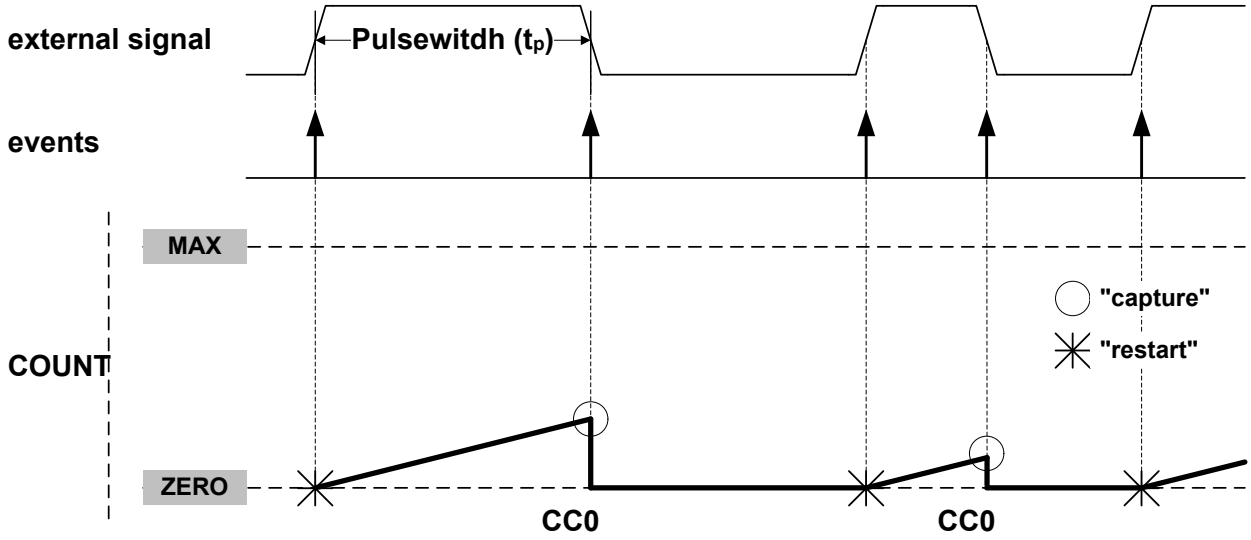
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx=1). If not, the capture action is ignored and the channel is enabled in compare mode of operation. Consequently, both channels must be enabled in order to fully characterize the input.

Pulse-Width Capture Action

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to DRVCTRL.INVEN or EVCTRL.TCEINV).

Figure 13-134. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

13.23.6.3 Additional Features

13.23.6.3.1 One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to zero.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT), and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

13.23.6.3.2 Time-Stamp Capture

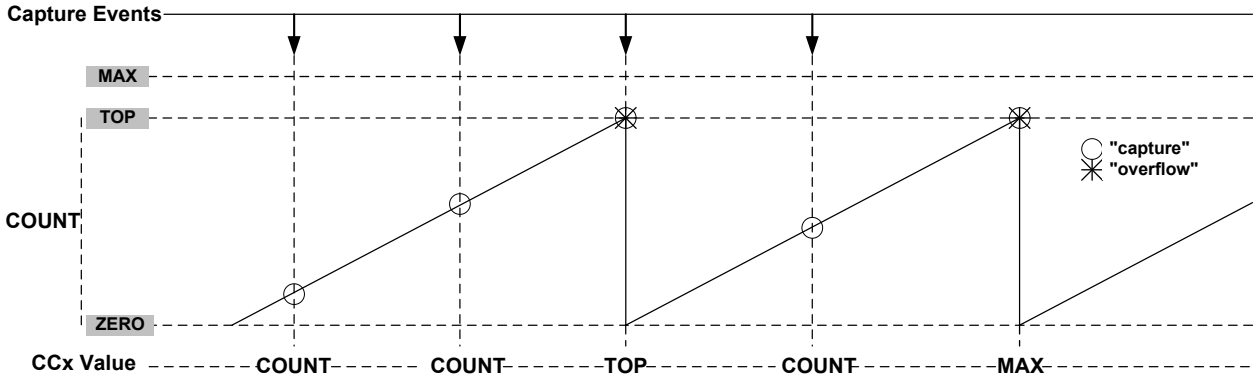
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 13-135. Time-Stamp



13.23.6.3.3 Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

CCx Content:

In CAPTMIN operations, CCx keeps the Minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMIN operation, capture is performed only when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum value has been detected.

13.23.6.3.4 Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

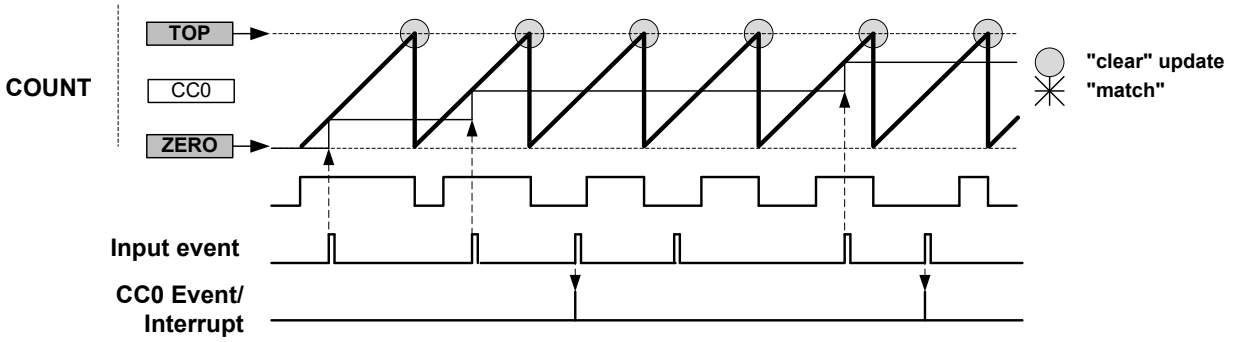
CCx Content:

In CAPTMAX operations, CCx keeps the Maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMAX operation, capture is performed only when on capture event time, the counter value is upper than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is lower or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Maximum value has been detected.

Figure 13-136. Maximum Capture Operation with CC0 Initialized with ZERO Value



13.23.6.4 DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

13.23.6.5 Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the TC is reset. See 13.23.7.1.7 INTFLAG for details on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.23.6.6 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)

- Count on event (COUNT)
- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

Related Links

13.18 EVSYS – Event System

13.23.6.7 Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

13.23.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

13.23.7 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.23.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.23.6.8 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.23.7.1 Register Summary - 8-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15 ... 0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E ... 0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							

13.23.7.1.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[13:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[31:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16

Value	Name	Description
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.23.7.1.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.1.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.1.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

13.23.7.1.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.1.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.1.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

13.23.7.1.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register. The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

13.23.7.1.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in . They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in .
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

Note:

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

13.23.7.1.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable
INVENx bit selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

13.23.7.1.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

13.23.7.1.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

13.23.7.1.13 Counter Value, 8-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – COUNT[7:0] Counter Value
These bits contain the current counter value.

13.23.7.1.14 Period Value, 8-bit Mode

Name: PER
Offset: 0x1B
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PER[7:0] Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

13.23.7.1.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CC[7:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

13.23.7.1.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PERBUF[7:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

13.23.7.1.17 Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CCBUF[7:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

13.23.7.2 Register Summary - 16-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
0x16 ... 0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x1E	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
0x20 ... 0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
0x32	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							

13.23.7.2.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[13:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[31:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16

Value	Name	Description
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.23.7.2.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.2.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.2.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

13.23.7.2.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.2.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.2.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

13.23.7.2.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register. The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

13.23.7.2.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in . They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in .
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

Note:

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

13.23.7.2.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable
INVENx bit selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

13.23.7.2.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

13.23.7.2.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

13.23.7.2.13 Counter Value, 16-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value
These bits contain the current counter value.

13.23.7.2.14 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

13.23.7.2.15 Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CCBUF[15:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

13.23.7.3 Register Summary - 32-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8					ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x18 ... 0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x20	CC1	7:0	CC[7:0]							
		15:8	CC[15:8]							
		23:16	CC[23:16]							
		31:24	CC[31:24]							
0x24 ... 0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							
0x34	CCBUF1	7:0	CCBUF[7:0]							
		15:8	CCBUF[15:8]							
		23:16	CCBUF[23:16]							
		31:24	CCBUF[31:24]							

13.23.7.3.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 20, 21 – COPENx Capture On Pin x Enable
Bit x of COPEN[13:0] selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable
Bit x of CAPTEN[31:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16

Value	Name	Description
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand
This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby
This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.23.7.3.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.3.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0] Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.23.7.3.4 Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

13.23.7.3.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.3.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.23.7.3.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

13.23.7.3.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register. The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

13.23.7.3.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in . They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in .
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

Note:

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

13.23.7.3.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – INVENx Output Waveform x Invert Enable
INVENx bit selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

13.23.7.3.11 Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

13.23.7.3.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 6, 7 – CCx Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.
This bit is set when the synchronization of PER between clock domains is started.
This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

13.23.7.3.13 Counter Value, 32-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value
These bits contain the current counter value.

13.23.7.3.14 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CC[31:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

13.23.7.3.15 Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUF[31:0] Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

13.24 TCC – Timer/Counter for Control Applications

13.24.1 Overview

The device provides three instances of the Timer/Counter for Control applications (TCC) peripheral, TCC[2:0].

Each TCC instance consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation such as frequency generation and pulse-width modulation.

Waveform extensions are intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. They allow for low- and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

Figure 13-137 shows all features in TCC.

Related Links
7.3.4 TCC Configurations

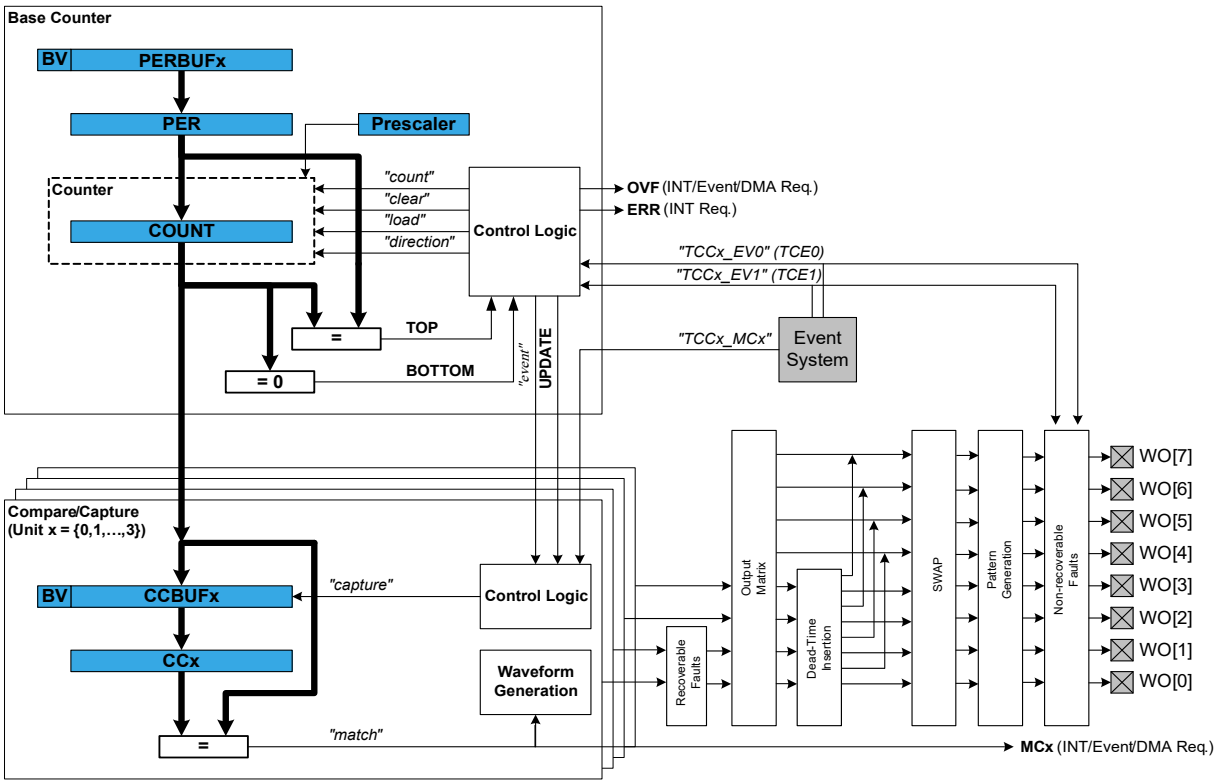
13.24.2 Features

- Up to four compare/capture channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel

  - Circular buffer on period and compare channel registers
- Waveform generation:
  - Frequency generation
  - Single-slope pulse-width modulation (PWM)
  - Dual-slope pulse-width modulation with half-cycle reload capability
- Input capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform extensions:
  - Configurable distribution of compare channels outputs across port pins
  - Low- and high-side output with programmable dead-time insertion
  - Waveform swap option with double buffer support
  - Pattern generation with double buffer support
  - Dithering support
- Fault protection for safe disabling of drivers:
  - Two recoverable fault sources
  - Two non-recoverable fault sources
  - Debugger can be source of non-recoverable fault
- Input events:
  - Two input events for counter
  - One input event for each channel
- Output events:
  - Three output events (Count, Re-Trigger and Overflow) available for counter
  - One Compare Match/Input Capture event output for each channel
- Interrupts:
  - Overflow and Re-Trigger interrupt
  - Compare Match/Input Capture interrupt
  - Interrupt on fault detection
- Can be used with DMA and can trigger DMA transactions

13.24.3 Block Diagram

Figure 13-137. Timer/Counter for Control Applications - Block Diagram



13.24.4 Signal Description

Pin Name	Type	Description
TCC/WO[0]	Digital output	Compare channel 0 waveform output
TCC/WO[1]	Digital output	Compare channel 1 waveform output
…	...	...
TCC/WO[WO_NUM-1]	Digital output	Compare channel n waveform output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

13.24.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.24.5.1 I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

13.17 PORT - I/O Pin Controller

13.24.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

13.24.5.3 Clocks

The TCC bus clock (CLK_TCCx_APB, with x instance number of the TCCx) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the generic clock controller before using the TCC. Note that TCC0 and TCC1 share a peripheral clock generator.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (CLK_TCCx_APB). Due to this asynchronicity, writing certain registers will require synchronization between the clock domains. Refer to 13.24.6.7 Synchronization for further details.

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.5 GCLK - Generic Clock Controller

13.24.5.4 DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.24.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.24.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

13.18 EVSYS – Event System

13.24.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to the 13.24.8.8 DBGCTRL register for details.

13.24.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture and Compare/Capture Buffer registers (CCx, CCBUFx)
- Control Waveform register (WAVE)
- Control Waveform Buffer register (WAVEBUF)
- Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTBUF)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.24.5.9 Analog Connections

Not applicable.

13.24.6 Functional Description

13.24.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 13-71. Timer/Counter for Control Applications - Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Generation Operations.
ZERO	The counter reaches ZERO when it contains all zeroes.
MAX	The counter reaches maximum when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g. counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

Each TCC instance has up to four compare/capture channels (CCx).

The counter register (COUNT), period registers with buffer (PER and PERBUF), and compare and capture registers with buffers (CCx and CCBUFx) are 16- or 24-bit registers, depending on each TCC instance. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate interrupt requests, request DMA transactions, or generate events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The MCE0 and MCE1 event sources are shared with the Recoverable Fault Unit. Only asynchronous events are used internally when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to *EVSYS – Event System*.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking, and qualification options. See also 13.24.6.3.5 Recoverable Faults.

In addition, six optional independent and successive units primarily intended for use with different types of motor control, ballast, LED, H-bridge, power converter, and other types of power switching applications, are implemented in some of TCC instances. See also Figure 13-137.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a safe and pre-configured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

Related Links

13.18 EVSYS – Event System

13.24.6.2 Basic Operation

13.24.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled(CTRLA.ENABLE=0):

- Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the “Enable-Protected” property in the register description.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Enable the TCC bus clock (CLK_TCCx_APB).
2. If Capture mode is required, enable the channel in capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESCSYNC).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR) to '1'.
4. Select the Waveform Generation operation in the WAVE register (WAVE.WAVEGEN).
5. Select the Waveform Output Polarity in the WAVE register (WAVE.POL).
6. The waveform output can be inverted for the individual channels using the Waveform Output Invert Enable bit group in the Driver register (DRVCTRL.INVEN).

13.24.6.2.2 Enabling, Disabling, and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a zero to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, will be reset to their initial state, and the TCC will be disabled. Refer to the Control A (13.24.8.1 CTRLA) register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

13.24.6.2.3 Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

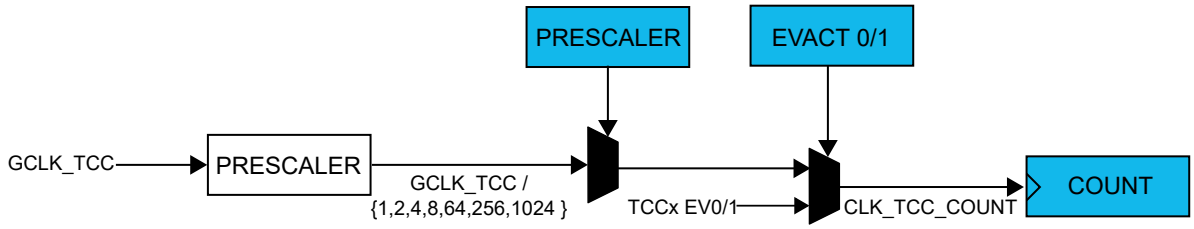
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCC clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCC_COUNT.

Figure 13-138. Prescaler



13.24.6.2.4 Counter Operation

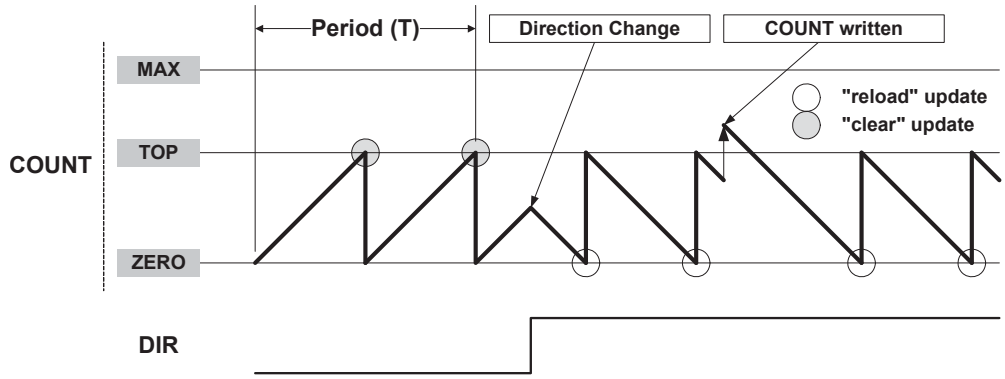
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it's counting up and one if counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it's counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

Figure 13-139. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. The COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also Figure 13-139.

Stop Command

A stop command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x2, STOP).

When a stop is detected while the counter is running, the counter will maintain its current value. If the waveform generation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State x Output Enable bit and Non- Recoverable State x Output Value bit in the Driver Control register (DRVCTRL.NREx and DRVCTRL.NRVx), and the Stop bit in the Status register is set (STATUS.STOP).

Pause Event Action

A pause command can be issued when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT1=0x3, STOP).

When a pause is detected, the counter will maintain its current value and all waveforms keep their current state, as long as a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0=0x3, START).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (INTFLAG.TRG). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:
When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Start Event Action

The start action can be selected in the Event Control register (EVCTRL.EVACT0=0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0=0x4, INC) and can change the counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1=0x4, DEC) and can change the counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

Related Links

13.24.6.3.1 One-Shot Operation

13.24.6.2.5 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to 13.24.6.2.6 Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously. The same condition generates a DMA request.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other waveforms generation modes, the update time occurs on counter wraparound, on overflow, underflow, or re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 13-72. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP(1) & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	–

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

Related Links

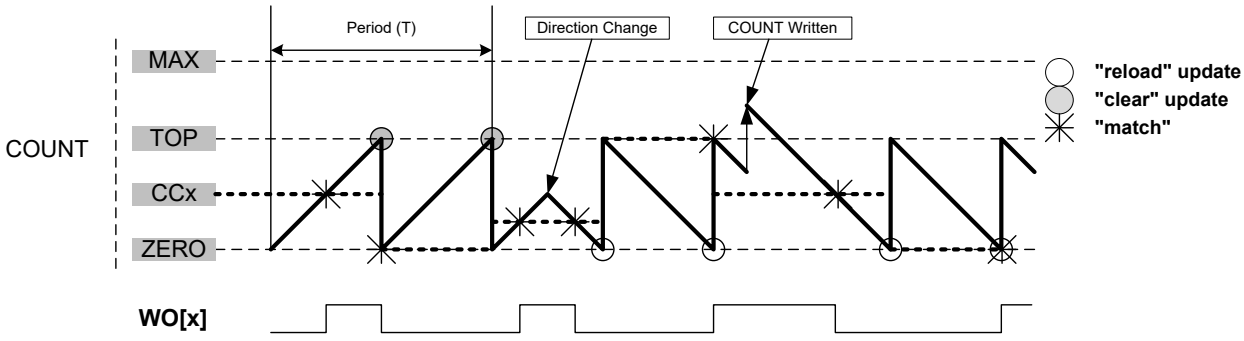
13.24.6.3.2 Circular Buffer

13.17 PORT - I/O Pin Controller

Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.

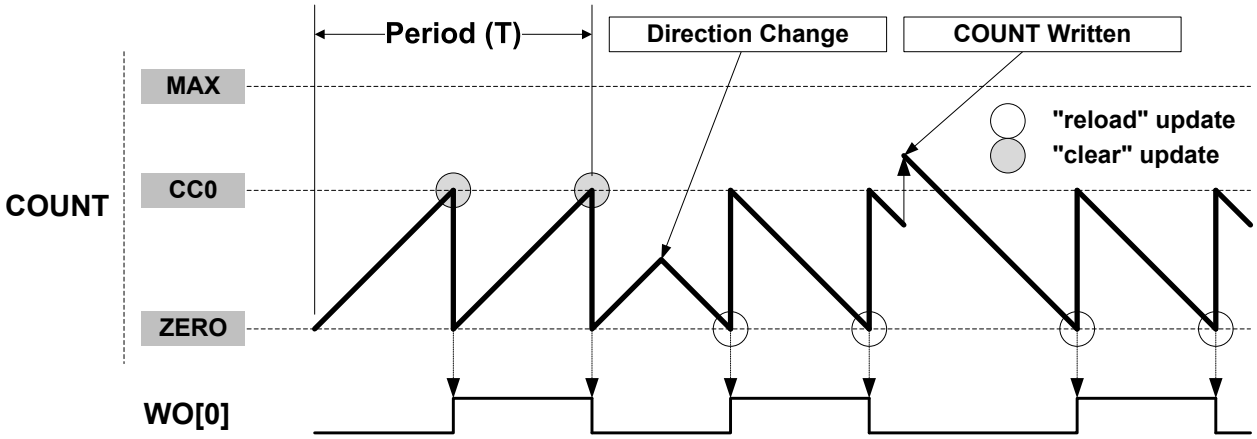
Figure 13-140. Normal Frequency Operation



Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 13-141. Match Frequency Operation



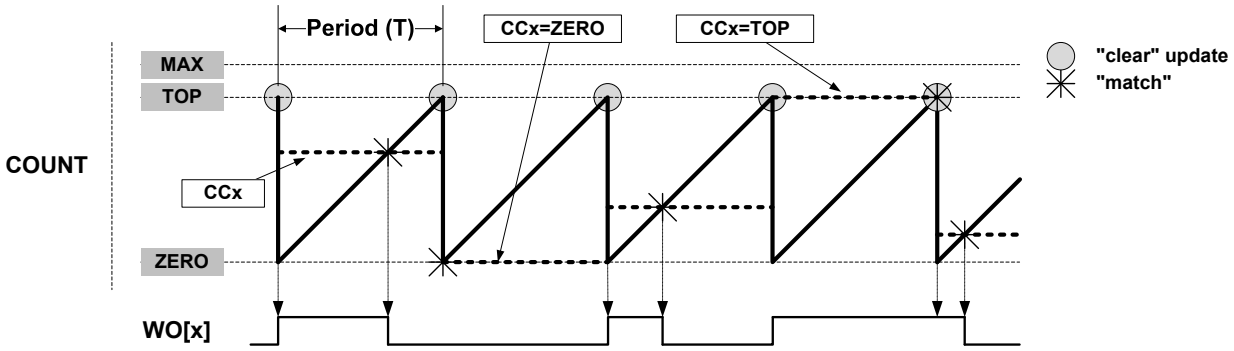
Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

Single-Slope PWM Operation

For single-slope PWM generation, the period time (T) is controlled by Top value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

Figure 13-142. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TCC}}{N(TOP+1)}$$

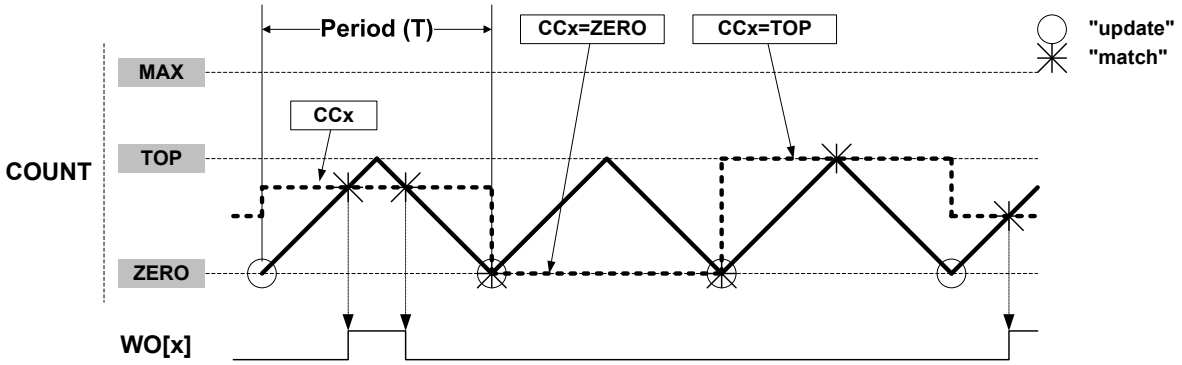
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Dual-Slope PWM Generation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCx control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set on compare match when up-counting, and cleared on compare match when down-counting. An interrupt/event is generated on TOP and/or ZERO, depend of Dual slope.

In DSBOTH operation, a second update time occurs on TOP when circular buffer is enabled.

Figure 13-143. Dual-Slope Pulse Width Modulation



Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM (R_{PWM_DS}):

$$R_{PWM\_DS} = \frac{\log(PER+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCC}, and can be calculated by the following equation:

$$f_{PWM\_DS} = \frac{f_{GCLK\_TCC}}{2N \cdot PER}$$

N represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCC}) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width (P_{PWM_DS}) depends on the compare channel (CCx) register value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$P_{PWM\_DS} = \frac{2N \cdot (TOP - CCx)}{f_{GCLK\_TCC}}$$

N represents the prescaler divider used.

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the CCx MSB bit defines the ramp on which the CCx Match interrupt or event is generated. (Rising if CCx[MSB]=0, falling if CCx[MSB]=1.)

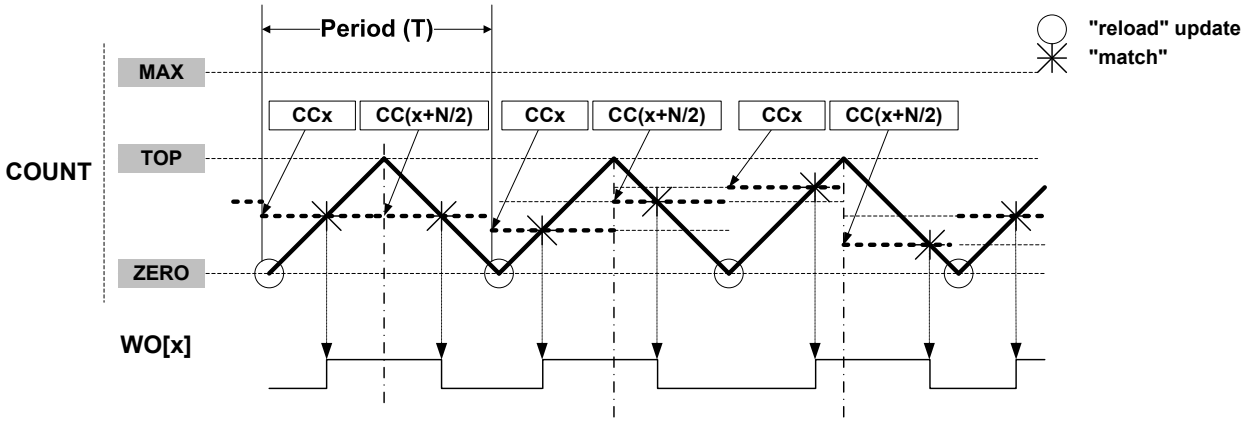
Related Links

13.24.6.3.2 Circular Buffer

Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time is controlled by PER while CCx control the generated waveform output edge during up-counting and CC(x+CC_NUM/2) control the generated waveform output edge during down-counting.

Figure 13-144. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 13-73. Waveform Generation Set/Clear Conditions

Waveform Generation operation	DIR	POLx	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CCx
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

13.24.6.2.6 Double Buffering

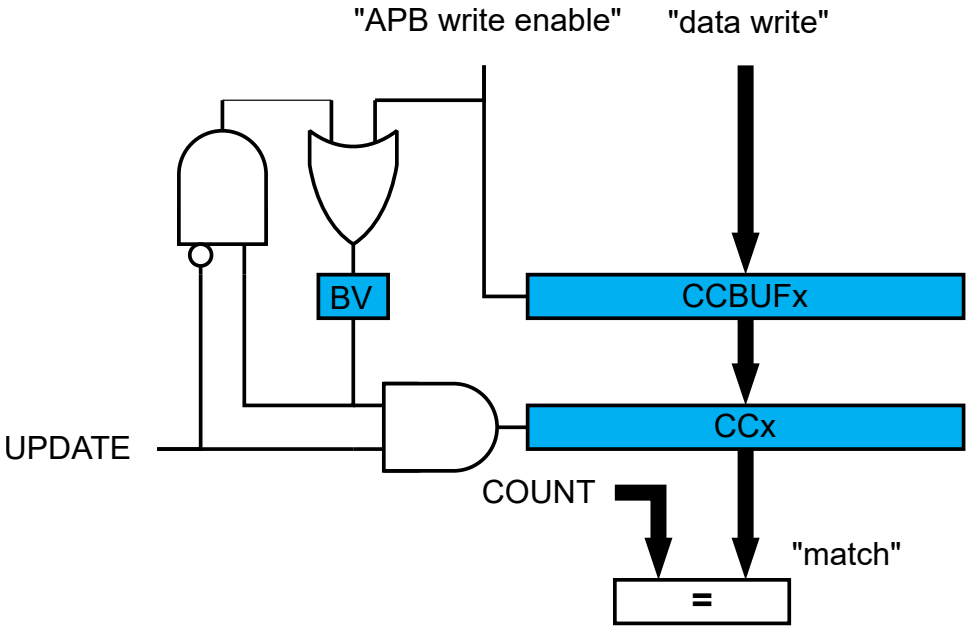
The Pattern (PATT), Waveform (WAVE), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV, WAVEBUFV, PERBUFV and CCBUFVx) bit in the STATUS register, which indicates that the Buffer register contains a valid value that can be copied into the corresponding register. As long as the respective Buffer Valid Status flag (PATTBUFV, WAVEBUFV,PERBUFV or CCBUFVx) are set to '1', the related SYNCBUSY bits are set (SYNCBUSY.PATT, SYNCBUSY.WAVE,SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PATT/PATTBUF, WAVE/WAVEBUF, PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and read access to the respective PATT, WAVE, PER or CCx register is invalid.

When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the Buffer Valid flags bit in the STATUS register are automatically cleared by hardware.

Note: Software update command (CTRLBSET.CMD=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

Figure 13-145. Compare Channel Double Buffering



Both the registers (PATT/WAVE/PER/CCx) and corresponding Buffer registers (PATTBUF/WAVEBUFV/PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD.

Note: In NFRQ, MFRQ or PWM Down-Counting Counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continuously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new Top value to the Period register (PER or CC0, depending on the Waveform Generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

Figure 13-146. Unbuffered Single-Slope Up-Counting Operation

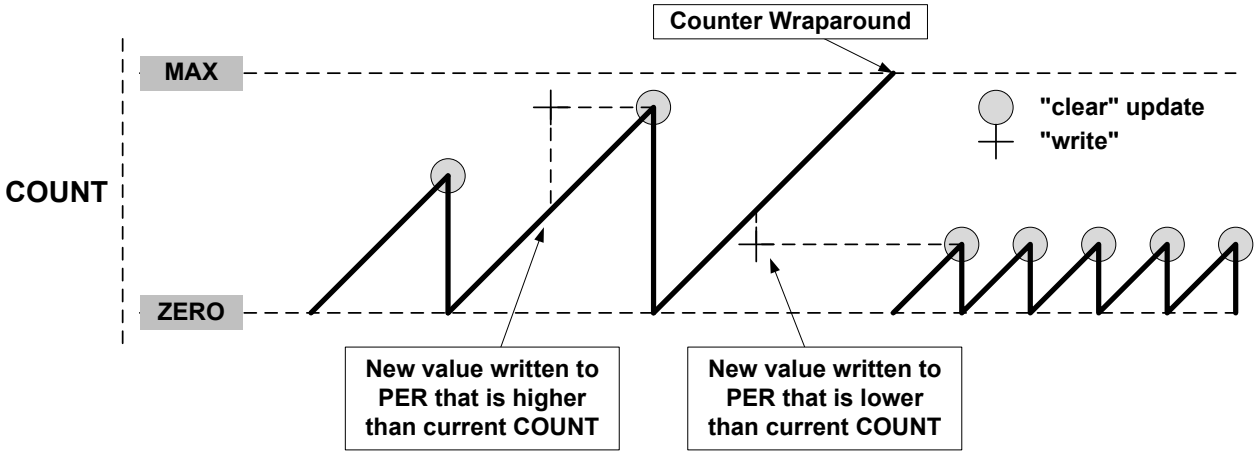
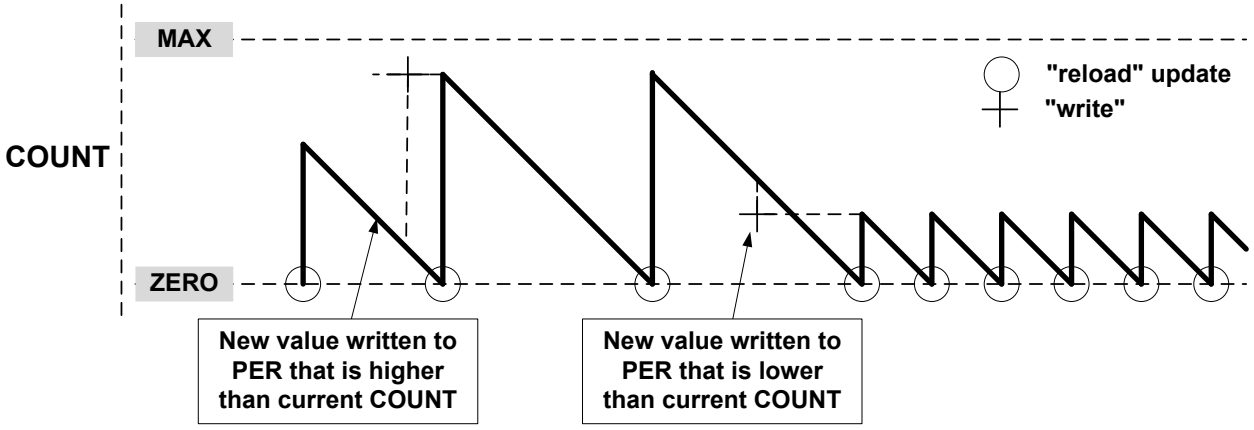
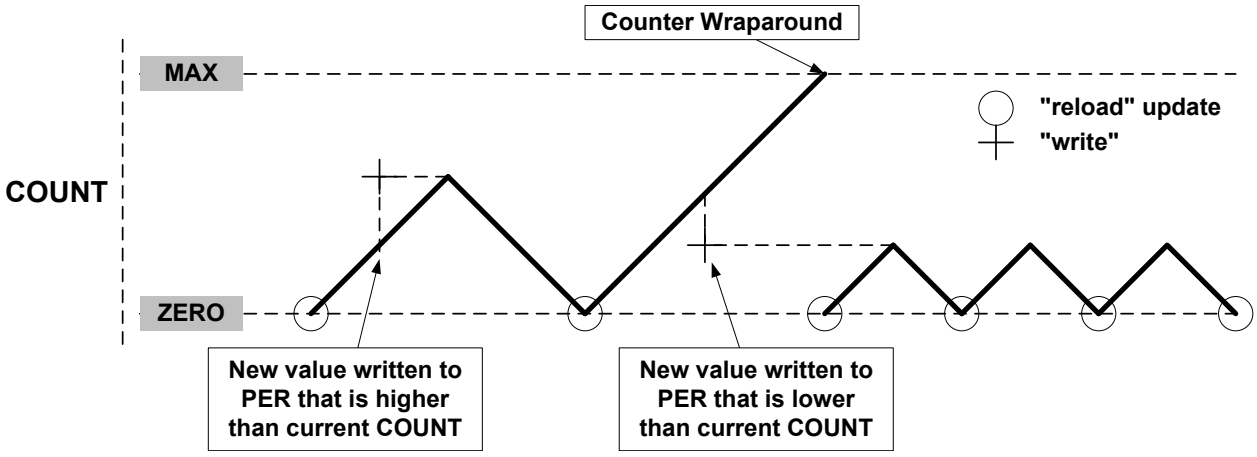


Figure 13-147. Unbuffered Single-Slope Down-Counting Operation



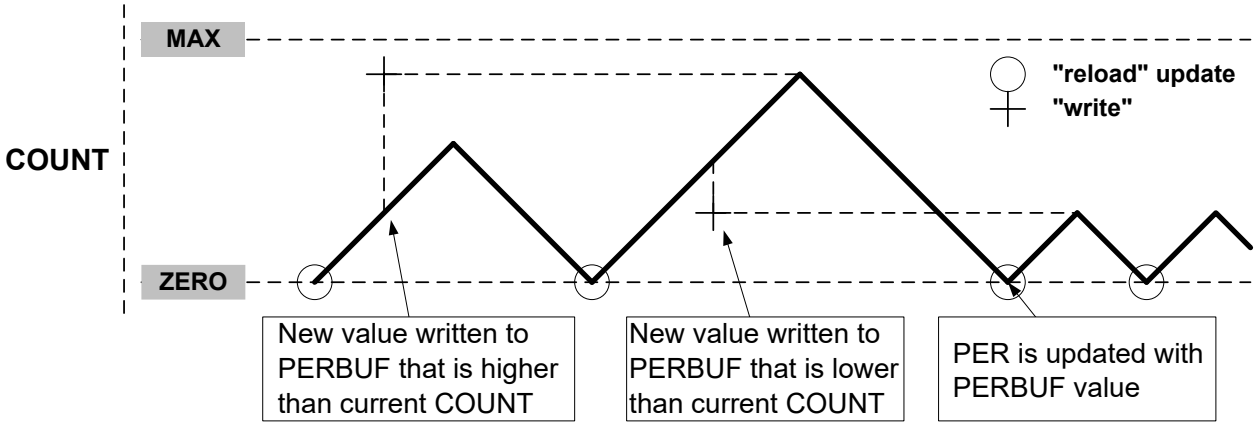
A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 13-146. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 13-148. Unbuffered Dual-Slope Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 13-149. This prevents wraparound and the generation of odd waveforms.

Figure 13-149. Changing the Period Using Buffering



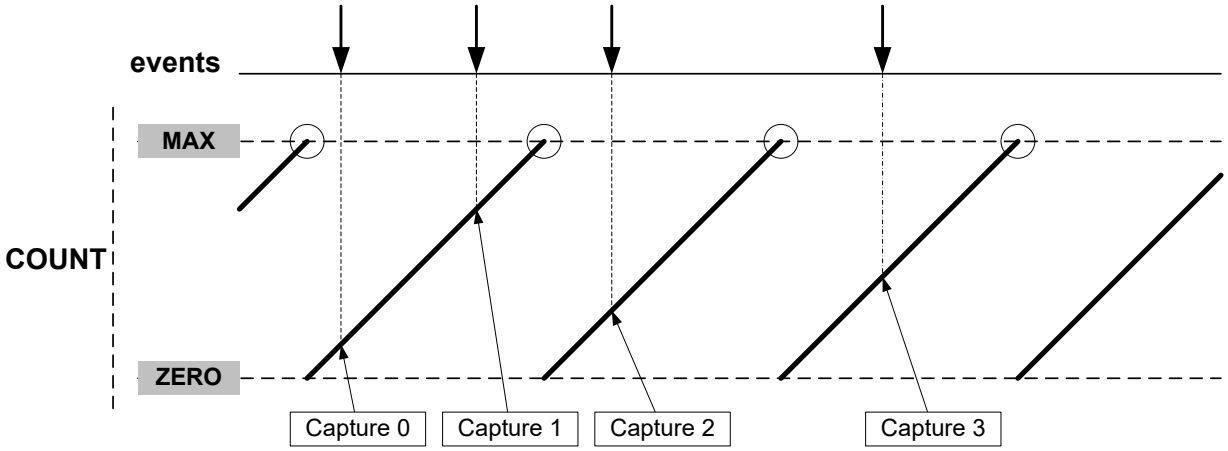
13.24.6.2.7 Capture Operations

To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

Event Capture Action

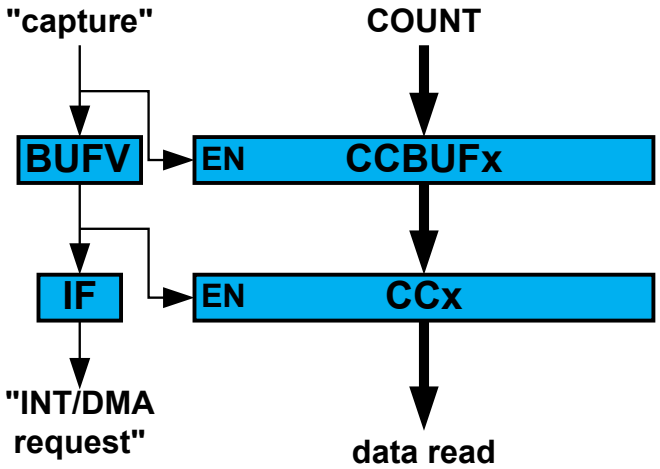
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 13-150. Input Capture Timing



For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBUFx register value can't be read, all captured data must be read from CCx register.

Figure 13-151. Capture Double Buffering



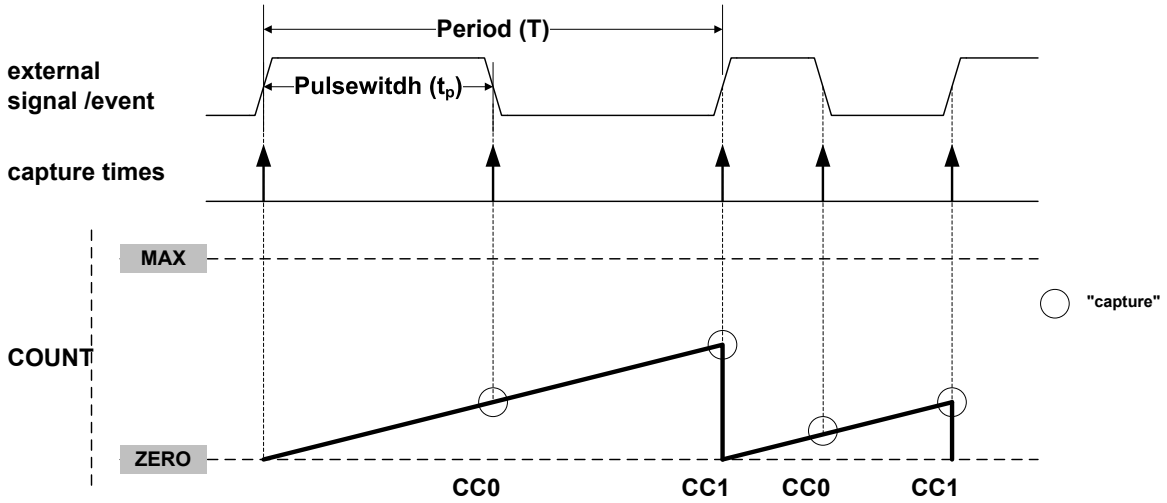
The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBUFV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and *dutyCycle* of an input signal:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 13-152. PWP Capture



Selecting PWP or PPW in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. When using PPW (period and pulse-width) event action, period *T* will be captured into CC0 and the pulse-width t_p into CC1. The PWP (Pulse-width and Period) event action offers the same functionality, but *T* will be captured into CC1 and t_p into CC0.

The Timer/Counter Event x Invert Enable bit in Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCEINVx=1, the wraparound will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in capture mode (CTRLA.CPTENx=1). If not, the capture action will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the INTFLAG.MCx is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: When up-counting (CTRLBSET.DIR=0), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLn.CAPTURE=CAPTMIN). To capture the full range including value 0, the TCC must be configured in down-counting mode (CTRLBSET.DIR=0).

Note: In dual-slope PWM operation, and when TOP is lower than MAX/2, the CCx MSB captures the CTRLB.DIR state to identify the ramp on which the capture has been done. For rising ramps CCx[MSB] is zero, for falling ramps CCx[MSB]=1.

13.24.6.3 Additional Features

13.24.6.3.1 One-Shot Operation

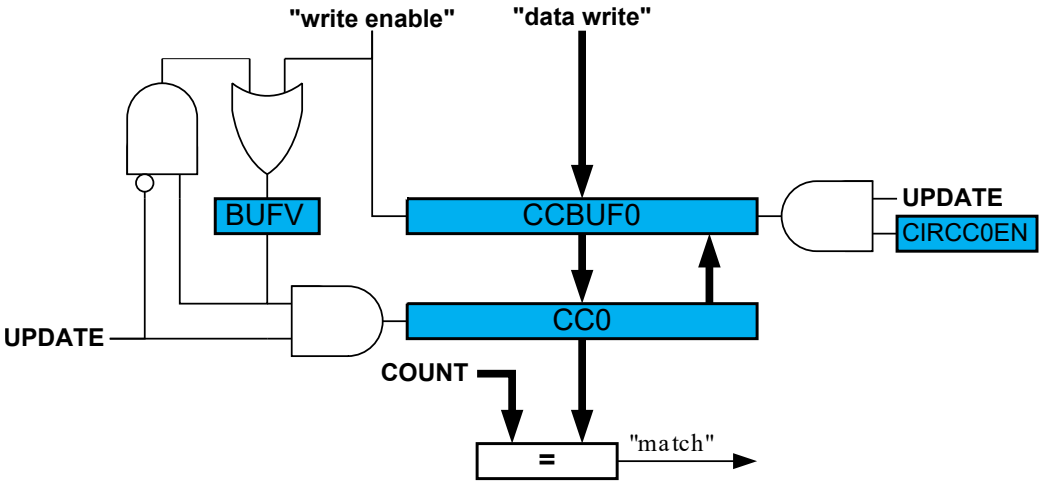
When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

13.24.6.3.2 Circular Buffer

The Period register (PER) and the compare channels register (CC0 to CC3) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 13-153. Circular Buffer on Channel 0



13.24.6.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
  value = cycle * dithercy;
  if (((MASK & value) + dithercy) > MASK)
    return 1;
 return 0;
}
```

Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

Note: If DITH4 mode is enabled, the last 4 significant bits from PER/CCx or COUNT register correspond to the DITHERCY value, rest of the bits corresponds to PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{32} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{64} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

Dithering on Pulse Width

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{16} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{32} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{64} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

Note: The PWM period will remain static in this case.

13.24.6.3.4 Ramp Operations

Three ramp operation modes are supported. All of them require the timer/counter running in single-slope PWM generation. The ramp mode is selected by writing to the Ramp Mode bits in the Waveform Control register (WAVE.RAMP).

RAMP1 Operation

This is the default PWM operation, described in Single-Slope PWM Generation.

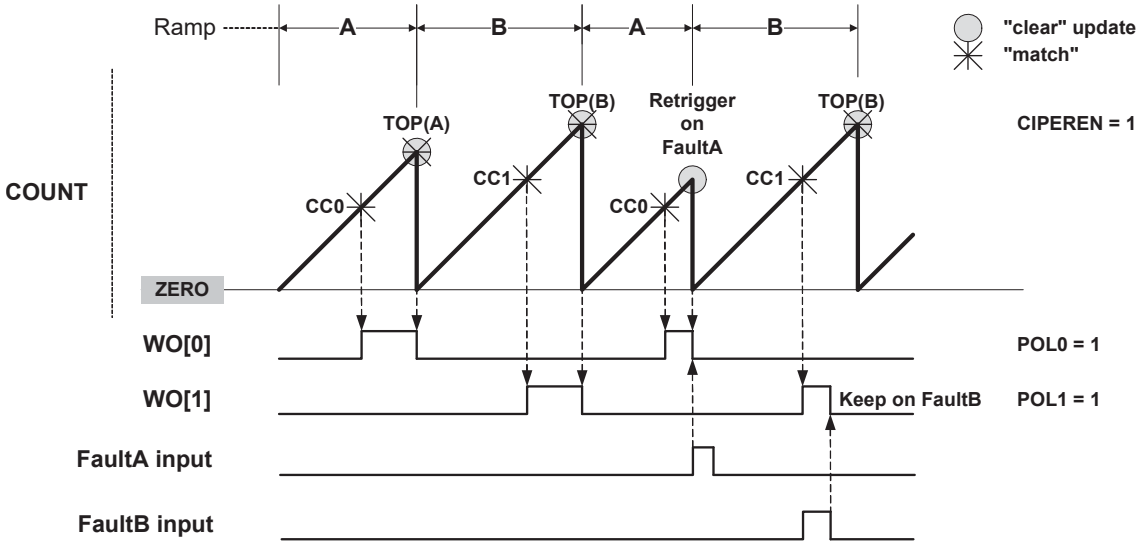
RAMP2 Operation

These operation modes are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see Figure 13-154. In cycle A, odd channel output is disabled, and in cycle B, even channel output is disabled. The ramp index changes after each update, but can be software modified using the Ramp index command bits in Control B Set register (CTRLBSET.IDXCMD).

Standard RAMP2 (RAMP2) Operation

Ramp A and B periods are controlled by the PER register value. The PER value can be different on each ramp by the Circular Period buffer option in the Wave register (WAVE.CIPEREN=1). This mode uses a two-channel TCC to generate two output signals, or one output signal with another CC channel enabled in capture mode.

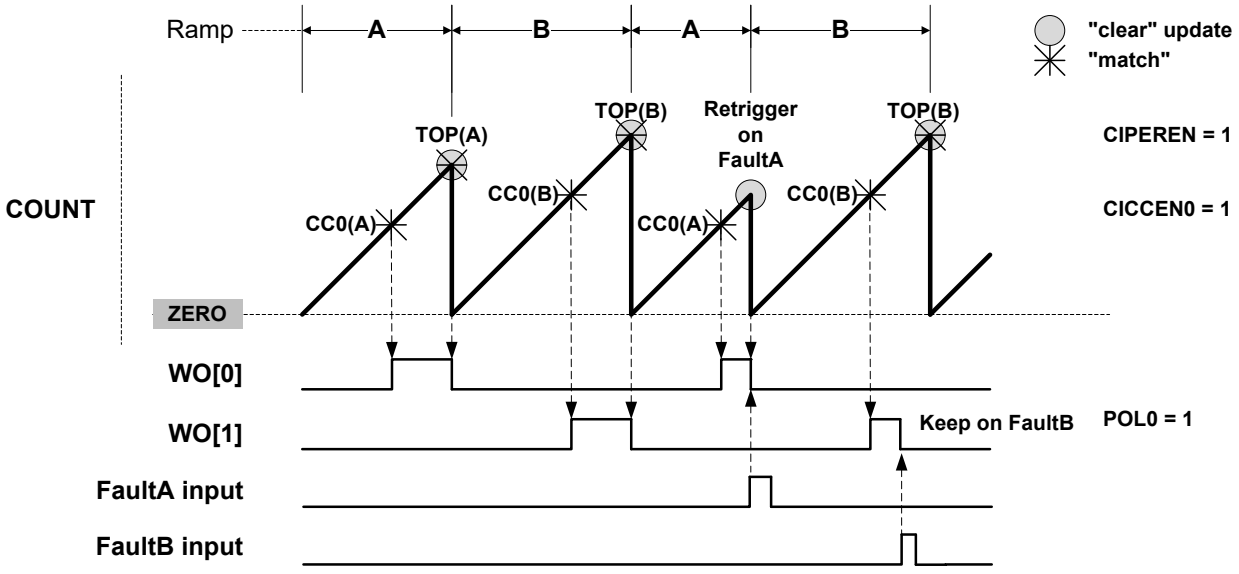
Figure 13-154. RAMP2 Standard Operation



Alternate RAMP2 (RAMP2A) Operation

Alternate RAMP2 operation is similar to RAMP2, but CC0 controls both WO[0] and WO[1] waveforms when the corresponding circular buffer option is enabled (CIPEREN=1). The waveform polarity is the same on both outputs. Channel 1 can be used in capture mode.

Figure 13-155. RAMP2 Alternate Operation



Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated to the use of circular buffers. In this mode, CC0 is controlling the period of ramp A and PER is controlling the period of ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of ramp A, if WAVE.POL0=0.

Figure 13-156. RAMP2 Critical Operation With More Than 2 Channels

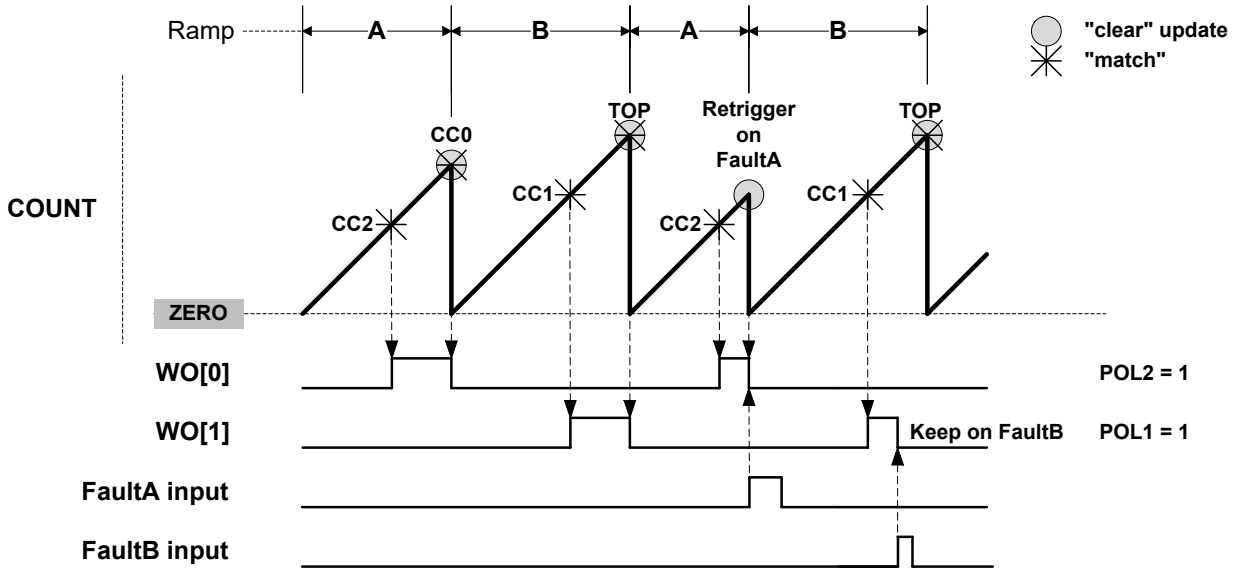
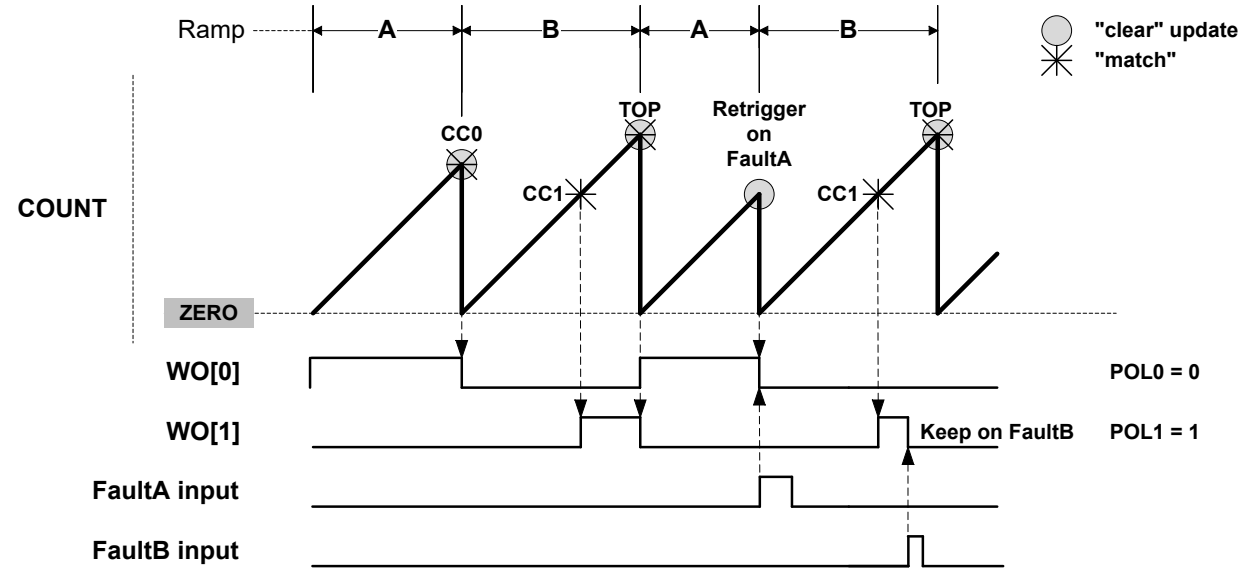


Figure 13-157. RAMP2 Critical Operation With 2 Channels



13.24.6.3.5 Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.

Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering

By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be delayed by FILTERVAL clock cycles.

Fault Blanking

This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL).
The blanking time $t_b$is calculated by

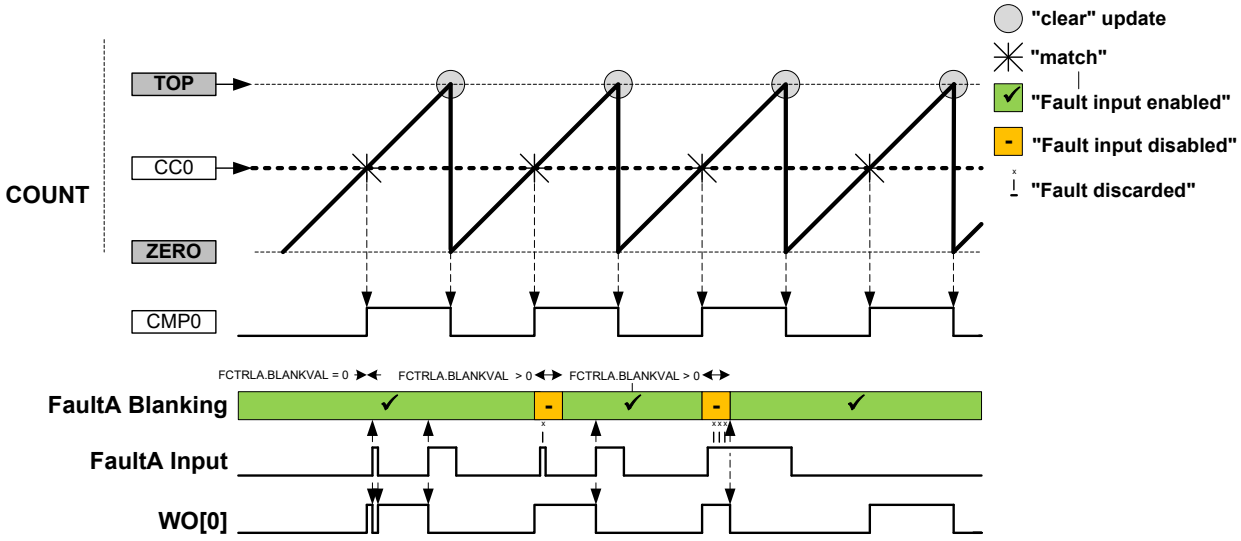
$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{GCLK\_TCCx\_PRESC}$ is the frequency of the prescaled peripheral clock frequency f_{GCLK_TCCx}.

The prescaler is enabled by writing '1' to the Fault n Blanking Prescaler bit (FCTRLn.BLANKPRESC). When disabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}. When enabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}/64.

The maximum blanking time (FCTRLn.BLANKVAL=
255) at f_{GCLK_TCCx}=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For f_{GCLK_TCCx}=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

Figure 13-158. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is enabled (FCTRLn.QUAL=1), the fault input is disabled all the time the corresponding channel output has an inactive level, as shown in the figures below.

Figure 13-159. Fault Qualification in RAMP1 Operation

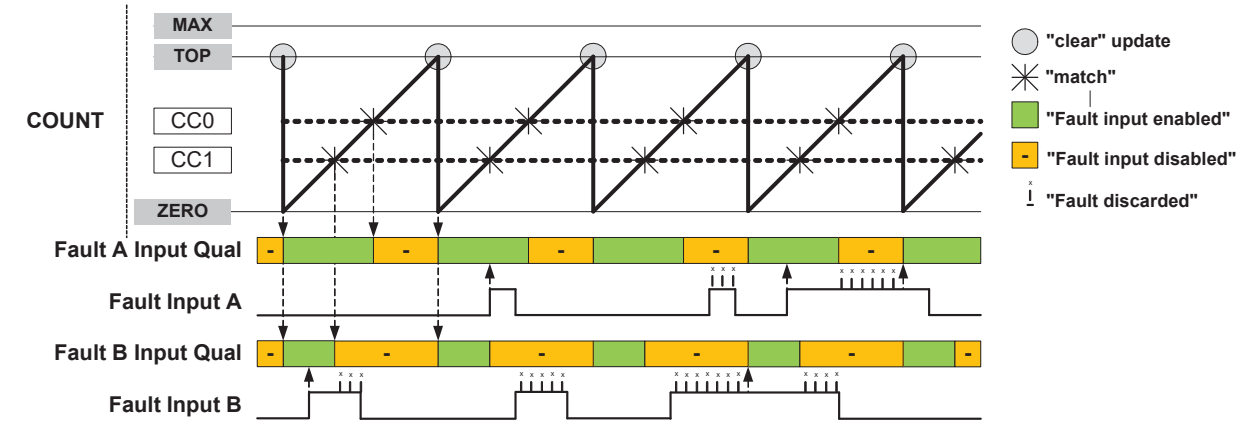
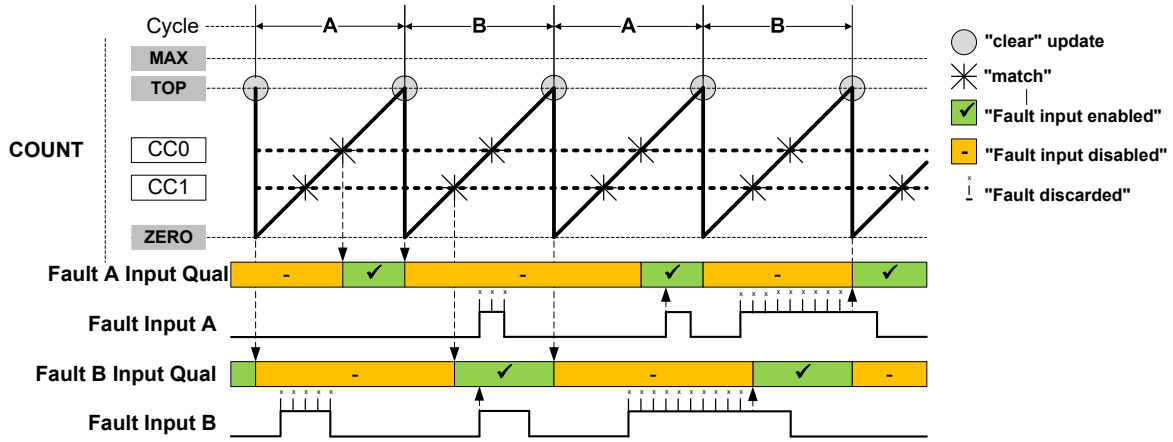


Figure 13-160. Fault Qualification in RAMP2 Operation with Inverted Polarity

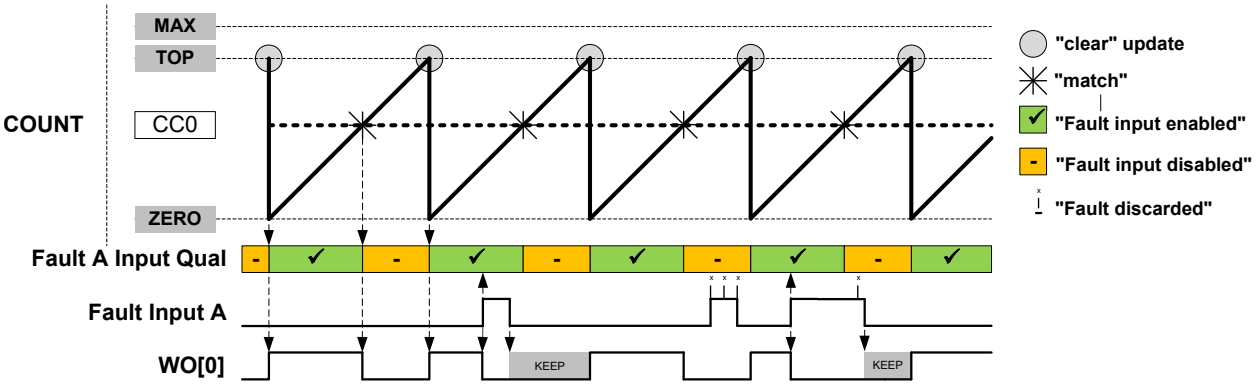


Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; hence two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

Keep Action This is enabled by writing the Fault n Keeper bit in the Recoverable Fault n Configuration register (FCTRLn.KEEP) to '1'. When enabled, the corresponding channel output will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.

Figure 13-161. Waveform Generation with Fault Qualification and Keep Action



Restart Action This is enabled by writing the Fault n Restart bit in Recoverable Fault n Configuration register (FCTRLn.RESTART) to '1'. When enabled, the timer/counter will be restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle, see Figure 13-162. In Ramp 1 mode, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

Note: For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see Figure 13-163. Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

Figure 13-162. Waveform Generation in RAMP1 mode with Restart Action

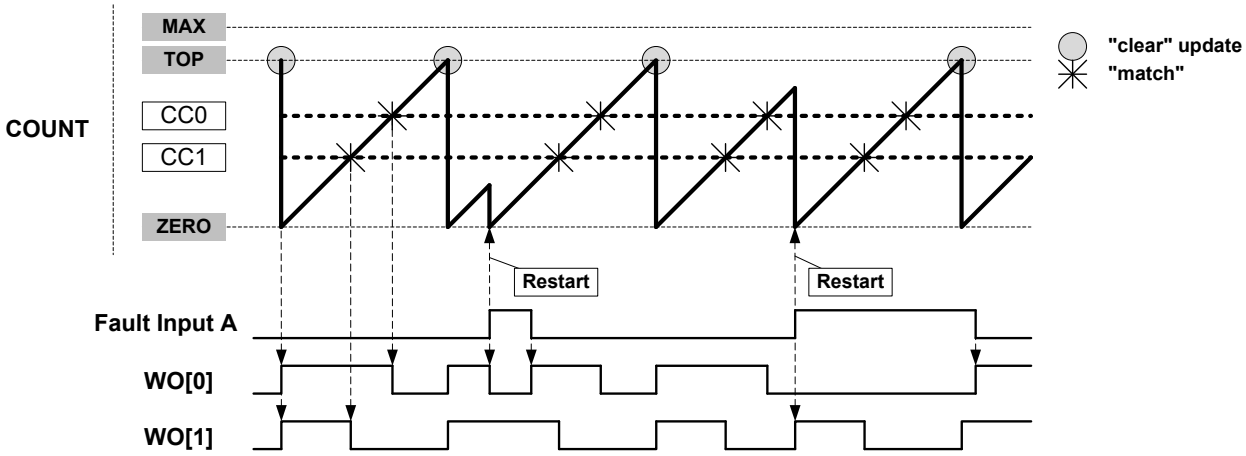
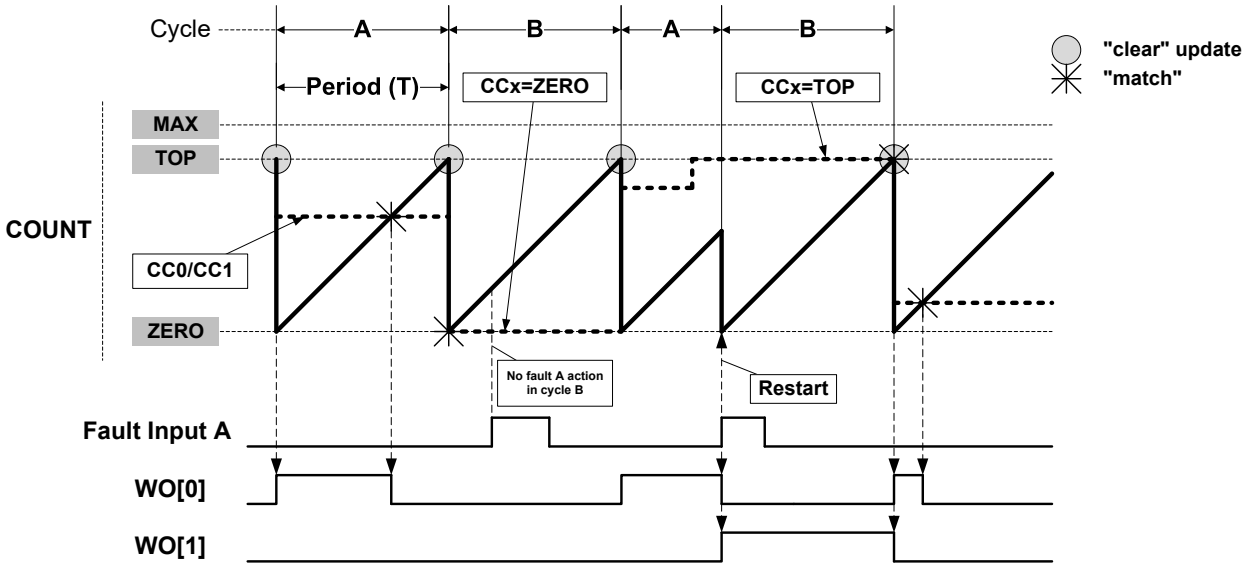


Figure 13-163. Waveform Generation in RAMP2 mode with Restart Action



Capture Action Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to 13.24.6.2.7 Capture Operations
- CAPTMIN - gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX - gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see Figure 13-164.

- LOCMIN - notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX - notifies by event or interrupt when a local maximum captured value is detected.
- DERIV0 - notifies by event or interrupt when a local extreme captured value is detected, see Figure 13-165.

CCx Content:

In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum captured values, see Figure 13-164. In LOCMIN, LOCMAX or DERIV0 operation, CCx follows the counter value at fault time, see Figure 13-165.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCx register value to a value different from zero (for CAPTMIN) top (for CAPTMAX). If the CCx register initial value is zero (for CAPTMIN) top (for CAPTMAX), no captures will be performed using the corresponding channel.

MCx Behaviour:

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCx interrupt flag is set only when the captured value is upper or equal (for LOCMIN) or lower or equal (for LOCMAX) to the previous captured value. So interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected. DERIV0 is equivalent to an OR function of (LOCMIN, LOCMAX).

In CAPT operation, capture is performed on each capture event. The MCx interrupt flag is set on each new capture.

In CAPTMIN and CAPTMAX operation, capture is performed only when on capture event time, the counter value is lower (for CAPTMIN) or upper (for CAPMAX) than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal (for CAPTMIN) or lower or equal (for CAPTMAX) to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected.

Interrupt Generation

In CAPT mode, an interrupt is generated on each filtered Fault n and each dedicated CCx channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 13-164. Capture Action “CAPTMAX”

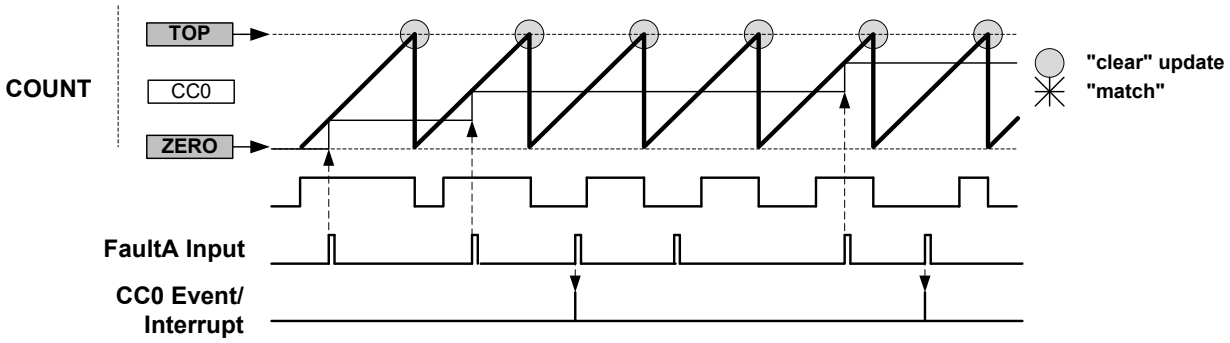
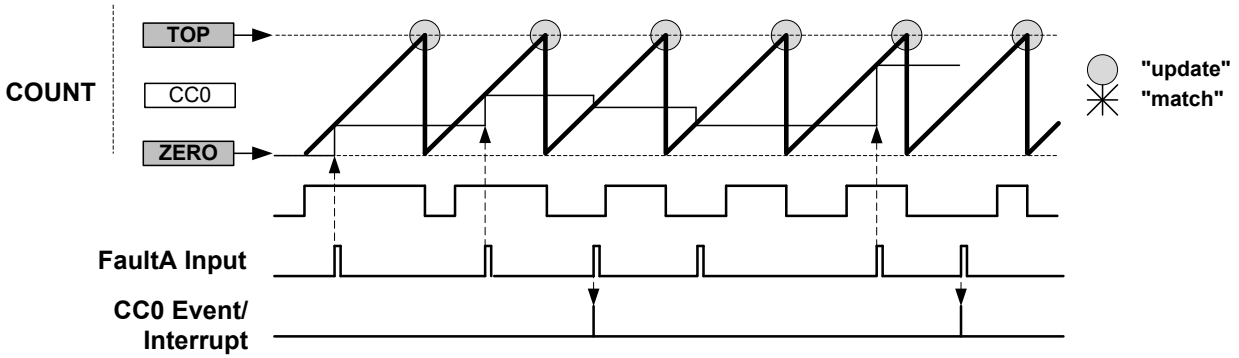


Figure 13-165. Capture Action “DERIV0”



Hardware Halt Action

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 13-166. Waveform Generation with Halt and Restart Actions

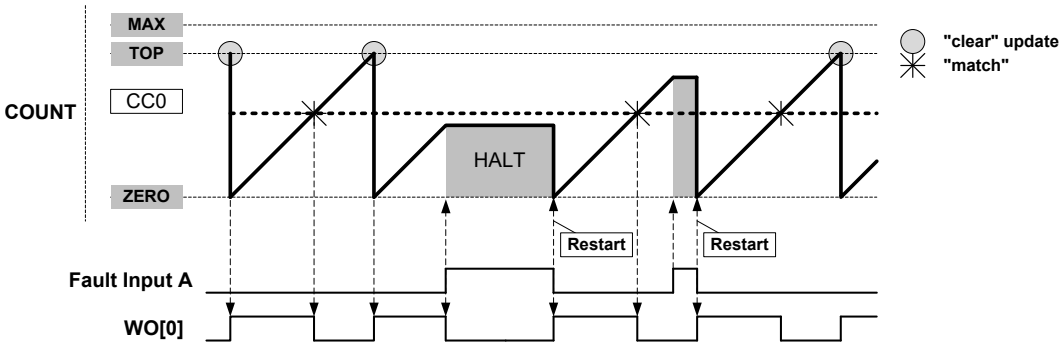
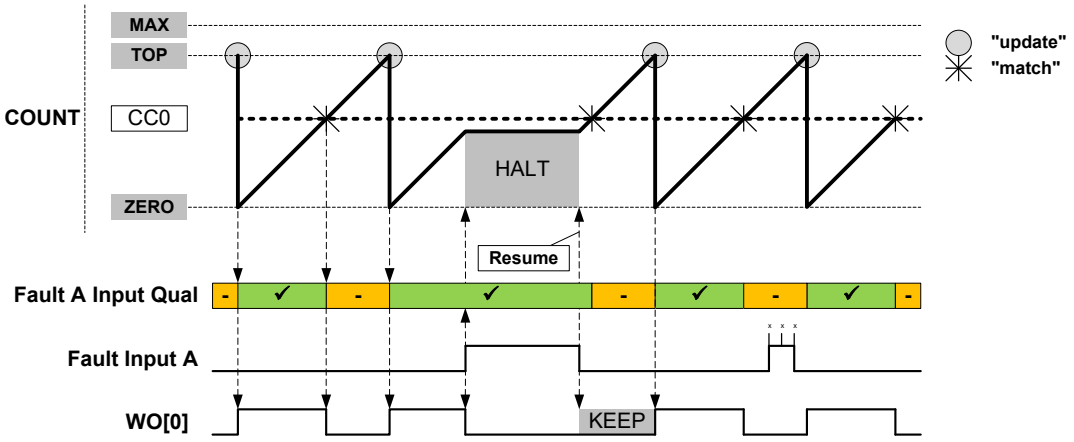


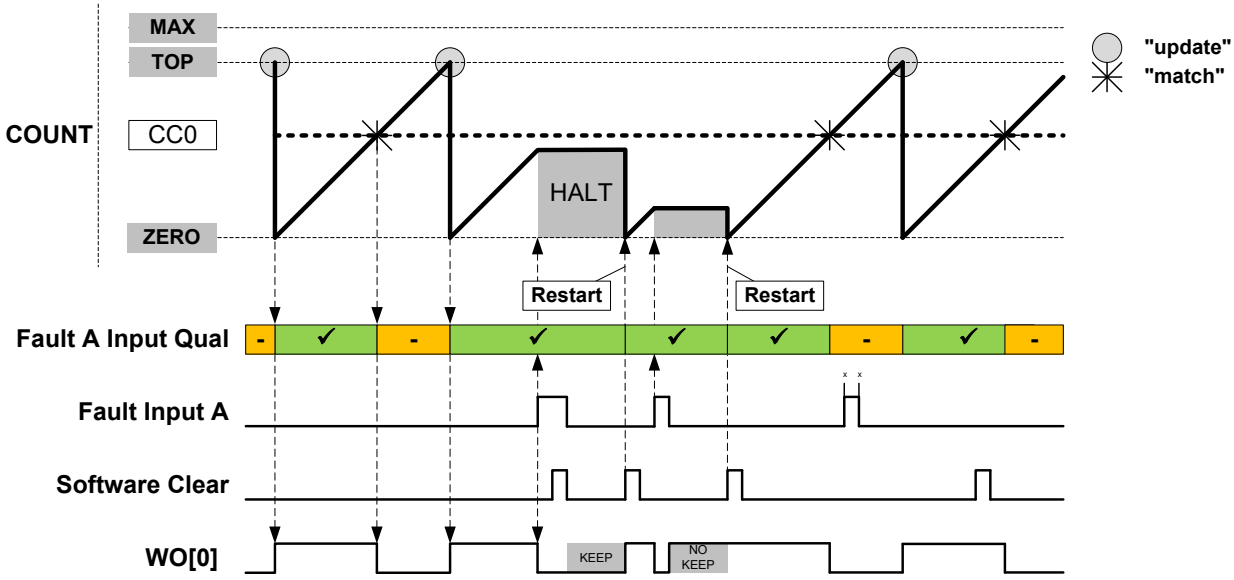
Figure 13-167. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action

This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT n bit in the STATUS register must be cleared by software.

Figure 13-168. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



13.24.6.3.6 Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register (DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD=1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

13.24.6.3.7 Time-Stamp Capture

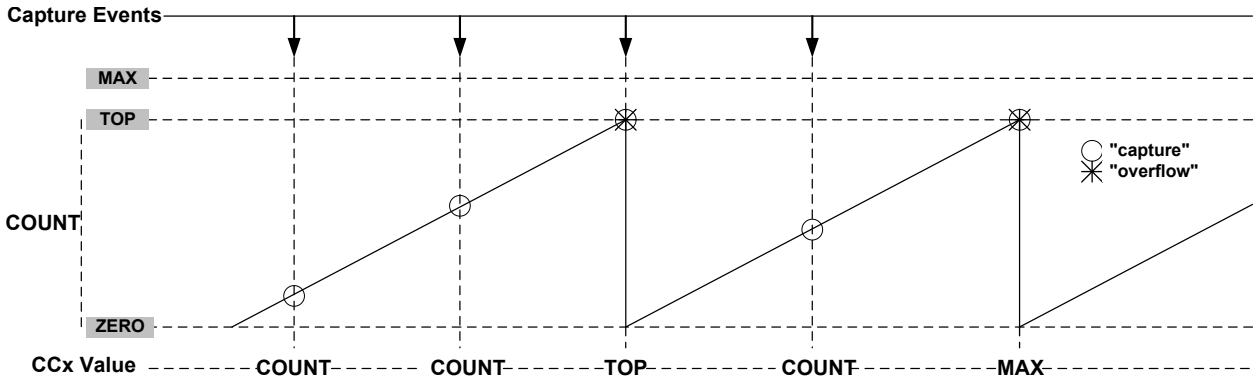
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 13-169. Time-Stamp



13.24.6.3.8 Waveform Extension

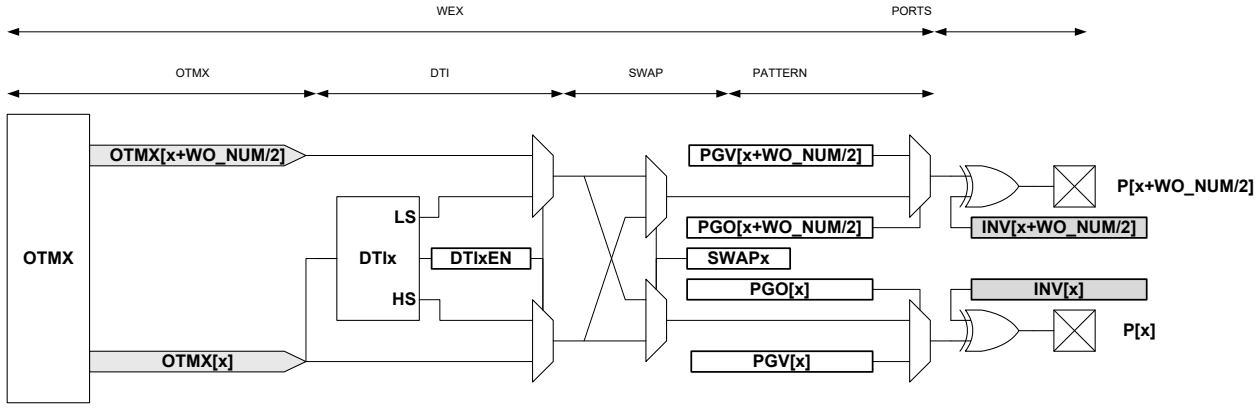
Figure 13-170 shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- Slice 0 DTI0 / SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1 / SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And more generally:

- Slice n DTIx / SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])

Figure 13-170. Waveform Extension Stage Details



The output matrix (OTMX) unit distributes compare channels, according to the selectable configurations in Table 13-74.

Table 13-74. Output Matrix Channel Pin Routing Configuration

Value	OTMX[x]							
0x0	CC3	CC2	CC1	CC0	CC3	CC2	CC1	CC0

.....continued								
Value	**OTMX[x]**							
0x1	CC1	CC0	CC1	CC0	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC1	CC1	CC0

Notes on Table 13-74:

- Configuration 0x0 is the default configuration. The channel location is the default one, and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0], and Channel 1 to OTMX[1]. If there are more outputs than channels, then channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
  Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings, with a boost stage.

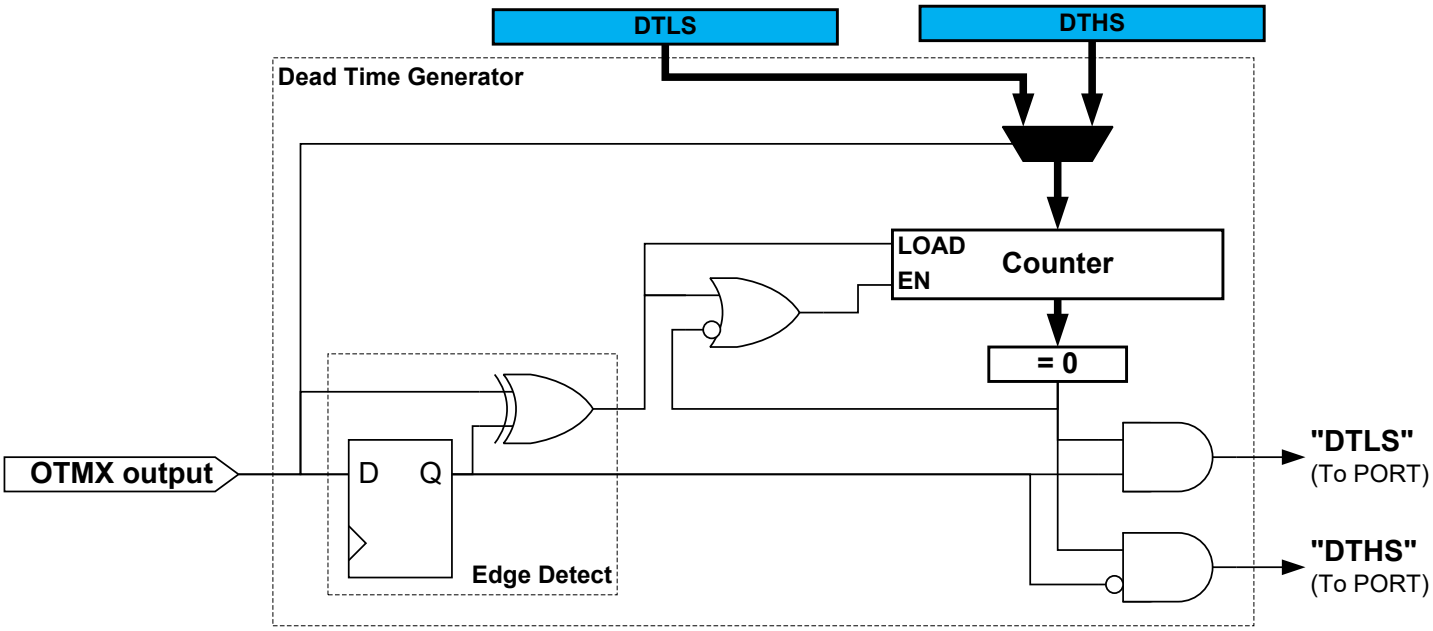
Table 13-75. Example: four compare channels on four outputs

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC0

The dead-time insertion (DTI) unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS will never switch simultaneously.

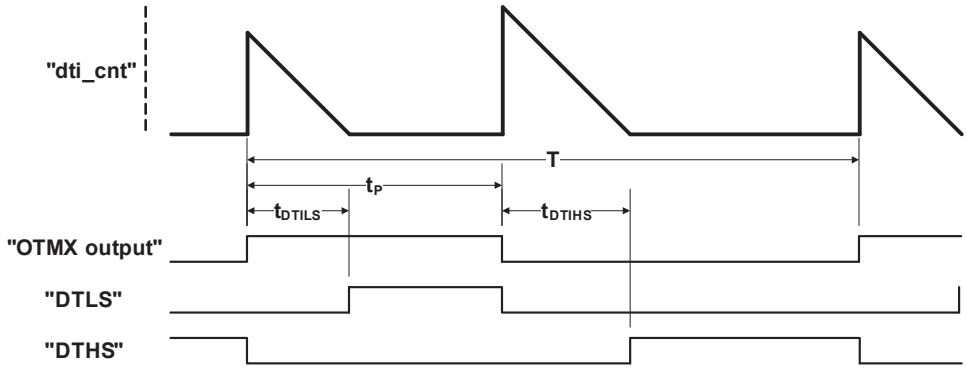
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. Figure 13-171 shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

Figure 13-171. Dead-Time Generator Block Diagram



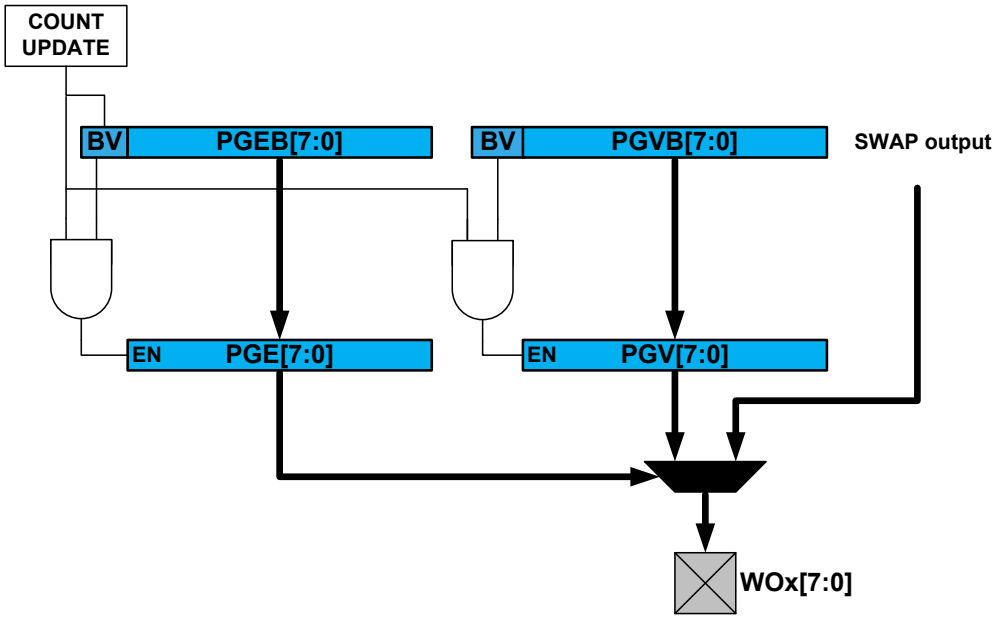
As shown in Figure 13-172, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge) it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge) it reloads the DTHS register.

Figure 13-172. Dead-Time Generator Timing Diagram



The pattern generator unit produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation sequence in brushless DC motors (BLDC), stepper motors, and full bridge control. See also Figure 13-173.

Figure 13-173. Pattern Generator Block Diagram



As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access directly the PATT.PGE, PATT.PGV bits registers.

13.24.6.4 Master/Slave Operation

Two TCC instances sharing the same GCLK_TCC clock, can be linked to provide more synchronized CC channels. The operation is enabled by setting the Master Synchronization bit in Control A register (CTRLA.MSYNC) in the Slave instance. When the bit is set, the slave TCC instance will synchronize the CC channels to the Master counter.

13.24.6.5 DMA, Interrupts, and Events

Table 13-76. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow / Underflow	Yes	Yes		Yes(1)	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes(2)	Yes(3)	For circular buffering: on DMA acknowledge For capture channel: when CCx register is read
Retrigger	Yes	Yes			
Count	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				
TCCx Event 0 input			Yes(4)		
TCCx Event 1 input			Yes(5)		

Notes:

1. DMA request set on overflow, underflow or re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In capture or circular modes.
4. On event input, either action can be executed:
   – re-trigger counter
   – control counter direction
   – stop the counter
   – decrement the counter
   – perform period and pulse width capture
   – generate non-recoverable fault
5. On event input, either action can be executed:
   – re-trigger counter
   – increment or decrement counter depending on direction
   – start the counter
   – increment or decrement counter based on direction
   – increment counter regardless of direction
   – generate non-recoverable fault

13.24.6.5.1 DMA Operation

The TCC can generate the following DMA requests:

Counter overflow (OVF)	If the Ones-shot Trigger mode in the control A register (CTRLA.DMAOS) is written to '0', the TCC generates a DMA request on each cycle when an update condition (overflow, underflow or re-trigger) is detected. When an update condition (overflow, underflow or re-trigger) is detected while CTRLA.DMAOS=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD=DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCx)	A DMA request is set only on a compare match if CTRLA.DMAOS=0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS=1, the DMA requests are not generated.
Channel Capture (MCx)	For a capture channel, the request is set when valid data is present in the CCx register, and cleared once the CCx register is read. In this operation mode, the CTRLA.DMAOS bit value is ignored.

DMA Operation with Circular Buffer

When circular buffer operation is enabled, the buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.

Note: Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

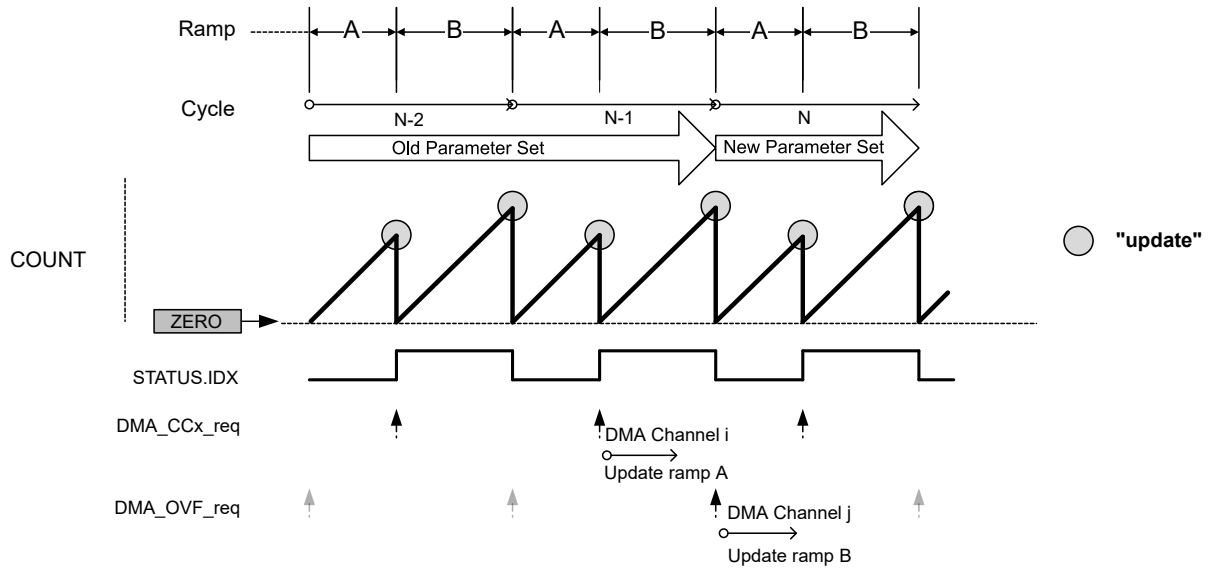
DMA Operation with Circular Buffer in RAMP and RAMP2A Mode

When a CCx channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of ramp A with an effective DMA transfer on previous ramp B (DMA acknowledge).

The update of all circular buffer values for ramp A can be done through a DMA channel triggered on a MC trigger. The update of all circular buffer values for ramp B, can be done through a second DMA channel triggered by the overflow DMA request.

Figure 13-174. DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled

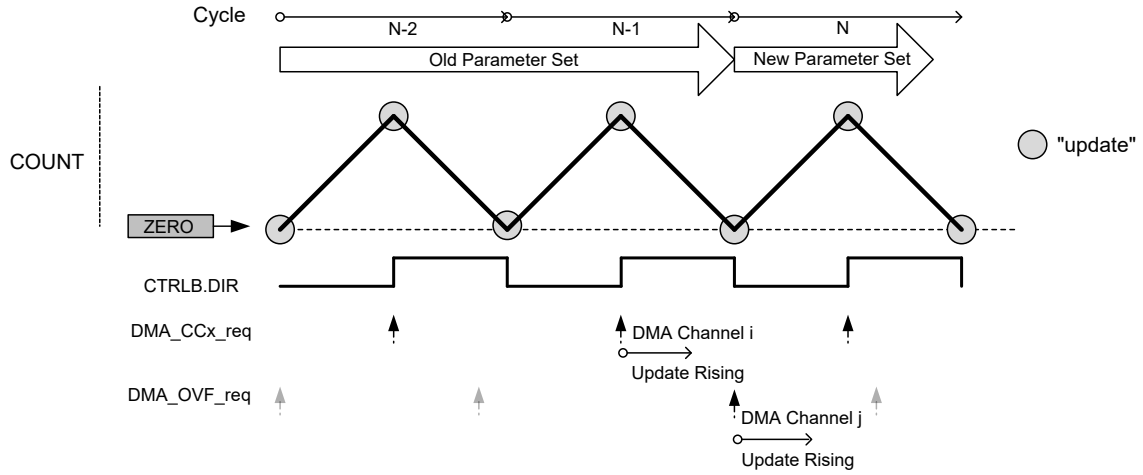


DMA Operation with Circular Buffer in DSBOTH Mode
When a CC channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 13-175. DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled



13.24.6.5.2 Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) – Refer also to the description of 13.24.8.9 EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)

- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/Sleep Mode Controller section for details.

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled or the TCC is reset. See 13.24.8.12 INTFLAG for details on how to clear Interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which Interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

13.24.6.5.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) – For further details, refer to the 13.24.8.9 EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Also, refer to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start – Start counting on the event rising edge. Further events will not restart the counter; the counter will keep on counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during the active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). For further details, refer to 13.24.8.9 EVCTRL.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on the input event.

Note: When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

Related Links

13.18 EVSYS – Event System

13.24.6.6 Sleep Mode Operation

The TCC can be configured to operate in any sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The MODULE can in any sleep mode wake up the device using interrupts or perform actions through the Event System.

13.24.6.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

13.4.3 Register Synchronization

13.24.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		RESOLUTION[1:0]					ENABLE	SWRST
		15:8	MSYNC	ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
		23:16	DMAOS							
		31:24					CPTEN3	CPTEN2	CPTEN1	CPTEN0
0x04	CTRLBCLR	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x06 ... 0x07	Reserved									
0x08	SYNCBUSY	7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
		15:8					CC3	CC2	CC1	CC0
		23:16								
		31:24								
0x0C	FCTRLA	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		23:16	BLANKVAL[7:0]							
		31:24					FILTERVAL[3:0]			
0x10	FCTRLB	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		23:16	BLANKVAL[7:0]							
		31:24					FILTERVAL[3:0]			
0x14	WEXCTRL	7:0							OTMX[1:0]	
		15:8					DTIENx	DTIENx	DTIENx	DTIENx
		23:16	DTLS[7:0]							
		31:24	DTHS[7:0]							
0x18	DRVCTRL	7:0	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
		15:8	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
		23:16	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
		31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
0x1C ... 0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
		15:8	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
		23:16					MCEI3	MCEI2	MCEI1	MCEI0
		31:24					MCEO3	MCEO2	MCEO1	MCEO0
0x24	INTENCLR	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16					MC3	MC2	MC1	MC0
		31:24								
0x28	INTENSET	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16					MC3	MC2	MC1	MC0
		31:24								
0x2C	INTFLAG	7:0					ERR	CNT	TRG	OVF
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		23:16					MC3	MC2	MC1	MC0
		31:24								
0x30	STATUS	7:0	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
		23:16					CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
		31:24					CMP3	CMP2	CMP1	CMP0

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	COUNT	7:0	COUNT[7:0]							
		15:8	COUNT[15:8]							
		23:16	COUNT[23:16]							
		31:24	COUNT[31:24]							
0x38	PATT	7:0	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
		15:8	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
0x3A ... 0x3B	Reserved									
0x3C	WAVE	7:0	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
		23:16					POL3	POL2	POL1	POL0
		31:24					SWAP3	SWAP2	SWAP1	SWAP0
0x40	PER	7:0	PER[1:0]		DITHER[5:0]					
		15:8	PER[9:2]							
		23:16	PER[17:10]							
		31:24	PER[25:18]							
0x44	CC0	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24	CC[25:18]							
0x48	CC1	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24	CC[25:18]							
0x4C	CC2	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24	CC[25:18]							
0x50	CC3	7:0	CC[1:0]		DITHER[5:0]					
		15:8	CC[9:2]							
		23:16	CC[17:10]							
		31:24	CC[25:18]							
0x54 ... 0x63	Reserved									
0x64	PATTBUF	7:0	PGEB7	PGEB6	PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
		15:8	PGVB7	PGVB6	PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
0x66 ... 0x67	Reserved									
0x68	WAVEBUF	7:0	CIPERENB		RAMPB[1:0]			WAVEGENB[2:0]		
		15:8					CICCENB3	CICCENB2	CICCENB1	CICCENB0
		23:16								
		31:24					SWAPB 3	SWAPB 2	SWAPB 1	SWAPB 0
0x6C	PERBUF	7:0	PERBUF[1:0]		DITHERBUF[5:0]					
		15:8	PERBUF[9:2]							
		23:16	PERBUF[17:10]							
		31:24	PERBUF[25:18]							
0x70	CCBUF0	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24	CCBUF[25:18]							
0x74	CCBUF1	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24	CCBUF[25:18]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	CCBUF2	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24	CCBUF[25:18]							
0x7C	CCBUF3	7:0	CCBUF[1:0]		DITHERBUF[5:0]					
		15:8	CCBUF[9:2]							
		23:16	CCBUF[17:10]							
		31:24	CCBUF[25:18]							

13.24.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.24.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized (ENABLE, SWRST)

Bit	31	30	29	28	27	26	25	24
					CPTEN3	CPTEN2	CPTEN1	CPTEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DMAOS							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	MSYNC	ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RESOLUTION[1:0]					ENABLE	SWRST
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bits 24, 25, 26, 27 – CPTENx Capture Channel x Enable
These bits are used to select the capture or compare operation on channel x.
Writing a '1' to CPTENx enables capture on channel x.
Writing a '0' to CPTENx disables capture on channel x.

Bit 23 – DMAOS DMA One-Shot Trigger Mode
This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit will generate a DMA trigger on TCC cycle following a TCC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit will generate DMA triggers on each TCC cycle.
This bit is not synchronized.

Bit 15 – MSYNC Master Synchronization (only for TCC slave instance)
This bit must be set if the TCC counting operation must be synchronized on its Master TCC.
This bit is not synchronized.

Value	Description
0	The TCC controls its own counter.
1	The counter is controlled by its Master TCC.

Bit 14 – ALOCK Auto Lock
This bit is not synchronized.

Value	Description
0	The Lock Update bit in the Control B register (CTRLB.LUPD) is not affected by overflow/underflow, and re-trigger events
1	CTRLB.LUPD is set to '1' on each overflow/underflow or re-trigger event.

Bits 13:12 – PRESCYNC[1:0] Prescaler and Counter Synchronization
These bits select if on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock, or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on re-trigger event.

These bits are not synchronized.

Value	Name	Description	
		Counter Reloaded	Prescaler
0x0	GCLK	Reload or reset Counter on next GCLK	-
0x1	PRESC	Reload or reset Counter on next prescaler clock	-
0x2	RESYNC	Reload or reset Counter on next GCLK	Reset prescaler counter
0x3	Reserved		

Bit 11 – RUNSTDBY Run in Standby
This bit is used to keep the TCC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TCC is halted in standby.
1	The TCC continues to run in standby.

Bits 10:8 – PRESCALER[2:0] Prescaler
These bits select the Counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TCC
0x1	DIV2	Prescaler: GCLK_TCC/2
0x2	DIV4	Prescaler: GCLK_TCC/4
0x3	DIV8	Prescaler: GCLK_TCC/8
0x4	DIV16	Prescaler: GCLK_TCC/16
0x5	DIV64	Prescaler: GCLK_TCC/64
0x6	DIV256	Prescaler: GCLK_TCC/256
0x7	DIV1024	Prescaler: GCLK_TCC/1024

Bits 6:5 – RESOLUTION[1:0] Dithering Resolution
These bits increase the TCC resolution by enabling the dithering options.
These bits are not synchronized.

Table 13-77. Dithering

Value	Name	Description
0x0	NONE	The dithering is disabled.
0x1	DITH4	Dithering is done every 16 PWM frames. PER[3:0] and CCx[3:0] contain dithering pattern selection.
0x2	DITH5	Dithering is done every 32 PWM frames. PER[4:0] and CCx[4:0] contain dithering pattern selection.
0x3	DITH6	Dithering is done every 64 PWM frames. PER[5:0] and CCx[5:0] contain dithering pattern selection.

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

13.24.8.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will read back zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.
Writing zero to this bit group has no effect.
Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Clear start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force COUNT read synchronization

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing zero to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	DISABLE	DISABLE Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT One-Shot
This bit controls one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop counting on the next overflow/underflow condition or on a stop command.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable the one-shot operation.

Value	Description
0	The TCC will update the counter value on overflow/underflow condition and continue operation.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.24.8.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBCLR) register.

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.
Writing zero to this bit group has no effect
Writing a valid value to this bit group will set the associated command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing a zero to these bits has no effect.
Writing a valid value to these bits will set a command.

Value	Name	Description
0x0	DISABLE	Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT One-Shot
This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the one-shot operation.

Value	Description
0	The TCC will count continuously.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will lock updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.

Bit 0 – DIR Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

13.24.8.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CC3	CC2	CC1	CC0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 8, 9, 10, 11 – CC Compare/Capture Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Channel x register between the clock domains is complete.
This bit is set when the synchronization of Compare/Capture Channel x register between clock domains is started.
CCx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.
This bit is set when the synchronization of CCx register between clock domains is started.

Bit 7 – PER PER Synchronization Busy
This bit is cleared when the synchronization of PER register between the clock domains is complete.
This bit is set when the synchronization of PER register between clock domains is started.

Bit 6 – WAVE WAVE Synchronization Busy
This bit is cleared when the synchronization of WAVE register between the clock domains is complete.
This bit is set when the synchronization of WAVE register between clock domains is started.

Bit 5 – PATT PATT Synchronization Busy
This bit is cleared when the synchronization of PATTERN register between the clock domains is complete.
This bit is set when the synchronization of PATTERN register between clock domains is started.

Bit 4 – COUNT COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT register between the clock domains is complete.
This bit is set when the synchronization of COUNT register between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS register between the clock domains is complete.
This bit is set when the synchronization of STATUS register between clock domains is started.

Bit 2 – CTRLB CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.
This bit is set when the synchronization of CTRLB register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

13.24.8.5 Fault Control A and B

Name: FCTRLx
Offset: 0x0C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					FILTERVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLANKVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0] Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault n Blanking Value Prescaler
This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is BLANKVAL* prescaled GCLK_TCC.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCC.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

Table 13-78. Fault n Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault n is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local minimum detection.

.....continued		
Value	**Name**	**Description**
0x3	CAPTMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection.

Bits 11:10 – CHSEL[1:0] Recoverable Fault n Capture Channel
These bits select the channel for capture operation triggered by recoverable Fault n.

Value	Name	Description
0x0	CC0	Capture value stored into CC0
0x1	CC1	Capture value stored into CC1
0x2	CC2	Capture value stored into CC2
0x3	CC3	Capture value stored into CC3

Bits 9:8 – HALT[1:0] Recoverable Fault n Halt Operation
These bits select the halt action for recoverable Fault n.

Value	Name	Description
0x0	DISABLE	Halt action disabled
0x1	HW	Hardware halt action
0x2	SW	Software halt action
0x3	NR	Non-recoverable fault

Bit 7 – RESTART Recoverable Fault n Restart
Setting this bit enables restart action for Fault n.

Value	Description
0	Fault n restart action is disabled.
1	Fault n restart action is enabled.

Bits 6:5 – BLANK[1:0] Recoverable Fault n Blanking Operation
These bits, select the blanking start point for recoverable Fault n.

Value	Name	Description
0x0	START	Blanking applied from start of the Ramp period
0x1	RISE	Blanking applied from rising edge of the waveform output
0x2	FALL	Blanking applied from falling edge of the waveform output
0x3	BOTH	Blanking applied from each toggle of the waveform output

Bit 4 – QUAL Recoverable Fault n Qualification
Setting this bit enables the recoverable Fault n input qualification.

Value	Description
0	The recoverable Fault n input is not disabled on CMPx value condition.
1	The recoverable Fault n input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP Recoverable Fault n Keep
Setting this bit enables the Fault n keep action.

Value	Description
0	The Fault n state is released as soon as the recoverable Fault n is released.
1	The Fault n state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Recoverable Fault n Source
These bits select the TCC event input for recoverable Fault n.

Event system channel connected to MCEx event input, must be configured to route the event asynchronously, when used as a recoverable Fault n input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCEx (x=0,1) event input
0x2	INVERT	Inverted MCEx (x=0,1) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

13.24.8.6 Waveform Extension Control

Name: WEXCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DTHS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DTLS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DTIENx	DTIENx	DTIENx	DTIENx
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							OTMX[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 31:24 – DTHS[7:0] Dead-Time High Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time high side.

Bits 23:16 – DTLS[7:0] Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time low side.

Bits 11,10,9,8 – DTIENx Dead-time Insertion Generator x Enable
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO_NUM/2], with the low side and high side waveform respectively.

Value	Description
0	No dead-time insertion override.
1	Dead time insertion override on signal outputs[x] and [x+WO_NUM/2], from matrix outputs[x] signal.

Bits 1:0 – OTMX[1:0] Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Table 13-74.

13.24.8.7 Driver Control

Name: DRVCTRL
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – FILTERVAL1[3:0] Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on TCE1 event input line. When the TCE1 event input line is configured as a synchronous event, this value must be 0x0.

Bits 27:24 – FILTERVAL0[3:0] Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on TCE0 event input line. When the TCE0 event input line is configured as a synchronous event, this value must be 0x0.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – INVENx Waveform Output x Inversion
These bits are used to select inversion on the output of channel x.
Writing a '1' to INVENx inverts output from WO[x].
Writing a '0' to INVENx disables inversion of output from WO[x].

Bits 8, 9, 10, 11, 12, 13, 14, 15 – NRVx NRVx Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – NREx Non-Recoverable State x Output Enable
These bits enable the override of individual outputs by NRVx value, under non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

13.24.8.8 Debug control

Name: DBGCTRL
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access						R/W		R/W
Reset						0		0

Bit 2 – FDDBD Fault Detection on Debug Break Detection
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.
By default this bit is zero, and the on-chip debug (OCD) fault protection is enabled. OCD break request from the OCD system will trigger non-recoverable fault. When this bit is set, OCD fault protection is disabled and OCD break request will not trigger a fault.

Value	Description
0	No faults are generated when TCC is halted in debug mode.
1	A non recoverable fault is generated and FAULTD flag is set when TCC is halted in debug mode.

Bit 0 – DBGRUN Debug Running State
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.

Value	Description
0	The TCC is halted when the device is halted in debug mode.
1	The TCC continues normal operation when the device is halted in debug mode.

13.24.8.9 Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					MCEO3	MCEO2	MCEO1	MCEO0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					MCEI3	MCEI2	MCEI1	MCEI0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27 – MCEOx Match or Capture Channel x Event Output Enable
These bits control if the match/capture event on channel x is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture x event is disabled and will not be generated.
1	Match/capture x event is enabled and will be generated for every compare/capture on channel x.

Bits 16, 17, 18, 19 – MCEIx Match or Capture Channel x Event Input Enable
These bits indicate if the match/capture x incoming event is enabled
These bits are used to enable match or capture input events to the CCx channel of TCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 14, 15 – TCEIx Timer/Counter Event Input x Enable
This bit is used to enable input event x to the TCC.

Value	Description
0	Incoming event x is disabled.
1	Incoming event x is enabled.

Bits 12, 13 – TCINVx Timer/Counter Event x Invert Enable
This bit inverts the event x input.

Value	Description
0	Input event source x is not inverted.
1	Input event source x is inverted.

Bit 10 – CNTEO Timer/Counter Event Output Enable
This bit is used to enable the counter cycle event. When enabled, an event will be generated on begin or end of counter cycle depending of CNTSEL[1:0] settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depend of CNTSEL[1:0] value.

Bit 9 – TRGEO Retrigger Event Output Enable
This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

Value	Description
0	Counter retrigger event is disabled and will not be generated.
1	Counter retrigger event is enabled and will be generated for every counter retrigger.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit is used to enable the overflow/underflow event. When enabled an event will be generated when the counter reaches the TOP or the ZERO value.

Value	Description
0	Overflow/underflow counter event is disabled and will not be generated.
1	Overflow/underflow counter event is enabled and will be generated for every counter overflow/ underflow.

Bits 7:6 – CNTSEL[1:0] Timer/Counter Interrupt and Event Output Selection
These bits define on which part of the counter cycle the counter event output is generated.

Value	Name	Description
0x0	BEGIN	An interrupt/event is generated at begin of each counter cycle
0x1	END	An interrupt/event is generated at end of each counter cycle
0x2	BETWEEN	An interrupt/event is generated between each counter cycle.
0x3	BOUNDARY	An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle.

Bits 5:3 – EVACT1[2:0] Timer/Counter Event Input 1 Action
These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TC on event
0x2	DIR (asynch)	Direction control
0x3	STOP	Stop TC on event
0x4	DEC	Decrement TC on event
0x5	PPW	Period captured into CC0 Pulse Width on CC1
0x6	PWP	Period captured into CC1 Pulse Width on CC0
0x7	FAULT	Non-recoverable Fault

Bits 2:0 – EVACT0[2:0] Timer/Counter Event Input 0 Action
These bits define the action the TCC will perform on TCE0 event input 0.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TC on event
0x4	INC	Increment TC on EVENT
0x5	COUNT (async)	Count on active state of asynchronous event
0x6	STAMP	Capture overflow times (Max value)
0x7	FAULT	Non-recoverable Fault

13.24.8.10 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 15 – FAULT1 Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT Counter Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.

Value	Description
1	The Overflow interrupt is enabled.

13.24.8.11 Interrupt Enable Set

Name: INTENSET
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – MCx Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 15 – FAULT1 Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Fault x Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which enables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT Counter Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

13.24.8.12 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – MCx Match or Capture Channel x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a match with the compare condition or once CCx register contain a valid capture value.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In Capture operation, this flag is automatically cleared when CCx register is read.

Bit 15 – FAULT1 Non-Recoverable Fault x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Non-Recoverable Fault x occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault x interrupt flag.

Bit 14 – FAULT0 Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt flag.

Bit 13 – FAULTB Recoverable Fault B Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 12 – FAULTA Recoverable Fault A Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 11 – DFS Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 10 – UFS Non-Recoverable Update Fault
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).
Writing a zero to this bit has no effect.
Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.

Bit 3 – ERR Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the error interrupt flag.

Bit 2 – CNT Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

13.24.8.13 Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: Read-Synchronized, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
					CMP3	CMP2	CMP1	CMP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
Access	R/W		R/W		R/W	R/W	R	R
Reset	0		0		0	0	0	1

Bits 24, 25, 26, 27 – CMPx Channel x Compare Value
This bit reflects the channel x output compare value.

Value	Description
0	Channel compare output value is 0.
1	Channel compare output value is 1.

Bits 16, 17, 18, 19 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel, this bit is set when a new value is written to the corresponding CCBUFx register. The bit is cleared either by writing a '1' to the corresponding location when CTRLB.LUPD is set, or automatically on an UPDATE condition.
For a capture channel, the bit is set when a valid capture value is stored in the CCBUFx register. The bit is automatically cleared when the CCx register is read.

Bits 14, 15 – FAULTx Non-recoverable Fault x State
This bit is set by hardware as soon as the non-recoverable Fault x condition occurs.
This bit is cleared by writing a '1' to this bit, and when the corresponding FAULTxIN status bit is low.
Once this bit is clear, the timer/counter will restart from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding STATEx bit. For further details on timer/counter commands, refer to the available commands description (13.24.8.3 CTRLBSET.CMD).

Bit 13 – FAULTB Recoverable Fault B State
This bit is set by hardware as soon as the recoverable Fault B condition occurs.
This bit can be cleared by hardware when the Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If the software halt command is enabled (FAULTB.HALT = SW), clearing this bit will release the timer/counter.

Bit 12 – FAULTA Recoverable Fault A State
This bit is set by the hardware as soon as the recoverable Fault A condition occurs.

This bit can be cleared by hardware when the Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If the software halt command is enabled (FAULTA.HALT = SW), clearing this bit will release the timer/counter.

Bit 11 – FAULT1IN Non-Recoverable Fault 1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

Bit 10 – FAULT0IN Non-Recoverable Fault 0 Input
This bit is set while an active Non-Recoverable Fault 0 input is present.

Bit 9 – FAULTBIN Recoverable Fault B Input
This bit is set while an active Recoverable Fault B input is present.

Bit 8 – FAULTAIN Recoverable Fault A Input
This bit is set while an active Recoverable Fault A input is present.

Bit 7 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. This bit is automatically cleared by hardware on the UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

Bit 5 – PATTBUFV Pattern Generator Value Buffer Valid
This bit is set when a new value is written to the PATTBUF register. This bit is automatically cleared by hardware on UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

Bit 3 – DFS Debug Fault State
This bit is set by hardware in Debug mode when DDBGCTRL.FDDBD bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in Debug mode.
When the bit is set, the counter is halted and the Waveforms state depend on the DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 2 – UFS Non-recoverable Update Fault State
This bit is set by hardware when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD). The bit is cleared by writing a ‘1’ to this bit.
When the bit is set, the waveforms state depend on the DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 1 – IDX Ramp Index
In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads ‘0’. For details on ramp operations, refer to 13.24.6.3.4 Ramp Operations.

Bit 0 – STOP Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT = 1).
This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

13.24.8.14 Counter Value

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value
These bits hold the value of the counter register.
Note: When the TCC is configured as 24- or 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0 (depicted)
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6

13.24.8.15 Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PGV Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGE Pattern Generation Output Enable
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' will override the corresponding SWAP output with the corresponding PGVn value.

13.24.8.16 Waveform

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
					SWAP3	SWAP2	SWAP1	SWAP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					POL3	POL2	POL1	POL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CICCEN3	CICCEN2	CICCEN1	CICCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bits 24, 25, 26, 27 – SWAP Swap DTI Output Pair x
Setting these bits enables output swap of DTI outputs [x] and [x+WO_NUM/2]. Note the DTIxEN settings will not affect the swap operation.

Bits 16, 17, 18, 19 – POL Channel Polarity x
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

Value	Name	Description
0	(single-slope PWM waveform generation)	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value
1	(single-slope PWM waveform generation)	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value.
0	(dual-slope PWM waveform generation)	Compare output is set to ~DIR when TCC counter matches CCx value
1	(dual-slope PWM waveform generation)	Compare output is set to DIR when TCC counter matches CCx value.

Bits 8, 9, 10, 11 – CICCEN Circular CC Enable x
Setting this bits enables the compare circular buffer option on channel. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.

Bit 7 – CIPEREN Circular Period Enable
Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERB register on UPDATE condition.

Bits 5:4 – RAMP[1:0] Ramp Operation
These bits select Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation

Value	Name	Description
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation

Bits 2:0 – WAVEGEN[2:0] Waveform Generation Operation
These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation should be used. These bits are not synchronized.

Value	Name	Description						
		Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVFIF/Event Up Down	
0x0	NFRQ	Normal Frequency	PER	TOP/Zero	Toggle	Stable	TOP	Zero
0x1	MFRQ	Match Frequency	CC0	TOP/Zero	Toggle	Stable	TOP	Zero
0x2	NPWM	Normal PWM	PER	TOP/Zero	Set	Clear	TOP	Zero
0x3	Reserved	–	–	–	–	–	–	–
0x4	DSCRITICAL	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x5	DSBOTTOM	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x6	DSBOTH	Dual-slope PWM	PER	TOP & Zero	~DIR	Stable	TOP	Zero
0x7	DSTOP	Dual-slope PWM	PER	Zero	~DIR	Stable	TOP	–

13.24.8.17 Period Value

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[1:0]		DITHER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:6 – PER[25:0] Period Value
These bits hold the value of the period register.
Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHER[5:0] Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse period every 64 PWM frames.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

13.24.8.18 Compare/Capture Channel x

Name: CCx
Offset: 0x44 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBUFx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Bit	31	30	29	28	27	26	25	24
	CC[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[1:0]		DITHER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:6 – CC[17:0] Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the m MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [23:m]
0x0 - NONE	23:0
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6 (depicted)

Bits 31:6 – CC[25:0] Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.
Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHER[5:0] Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse width every 64 PWM frames.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

13.24.8.19 Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGVB7	PGVB6	PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGEB7	PGEB6	PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PGVB Pattern Generation Output Value Buffer
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGV register on an UPDATE condition.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGEB Pattern Generation Output Enable Buffer
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGE register at an UPDATE condition.

13.24.8.20 Waveform Buffer

Name: WAVEBUF
Offset: 0x68
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
					SWAPB 3	SWAPB 2	SWAPB 1	SWAPB 0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CICCENB3	CICCENB2	CICCENB1	CICCENB0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPERENB		RAMPB[1:0]			WAVEGENB[2:0]		
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bits 24, 25, 26, 27 – SWAPB Swap DTI output pair x Buffer
These register bits are the buffer bits for the SWAP register bits. If double buffering is used, valid content in these bits is copied to the corresponding SWAPx bits on an UPDATE condition.

Bits 8, 9, 10, 11 – CICCENB Circular CCx Buffer Enable
These register bits are the buffer bits for CICCENx register bits. If double buffering is used, valid content in these bits is copied to the corresponding CICCENx bits on a UPDATE condition.

Bit 7 – CIPERENB Circular Period Enable Buffer
This register bit is the buffer bit for CIPEREN register bit. If double buffering is used, valid content in this bit is copied to the corresponding CIPEREN bit on a UPDATE condition.

Bits 5:4 – RAMPB[1:0] Ramp Operation Buffer
These register bits are the buffer bits for RAMP register bits. If double buffering is used, valid content in these bits is copied to the corresponding RAMP bits on a UPDATE condition.

Bits 2:0 – WAVEGENB[2:0] Waveform Generation Operation Buffer
These register bits are the buffer bits for WAVEGEN register bits. If double buffering is used, valid content in these bits is copied to the corresponding WAVEGEN bits on a UPDATE condition.

13.24.8.21 Period Buffer Value

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PERBUF[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERBUF[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:6 – PERBUF[25:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.
Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enabled, the value of this bit field is copied to the PER.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

13.24.8.22 Channel x Compare/Capture Buffer Value

Name: CCBUF
Offset: 0x70 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

CCBUFx is copied into CCx at TCC update time

Bit	31	30	29	28	27	26	25	24
	CCBUF[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:6 – CCBUF[25:0] Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBUFVx status bit.
Note: When the TCC is configured as 16- or 24-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.
Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

13.25 USB – Universal Serial Bus

13.25.1 Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting both device and embedded host modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The USB host mode supports up to 8 pipes. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

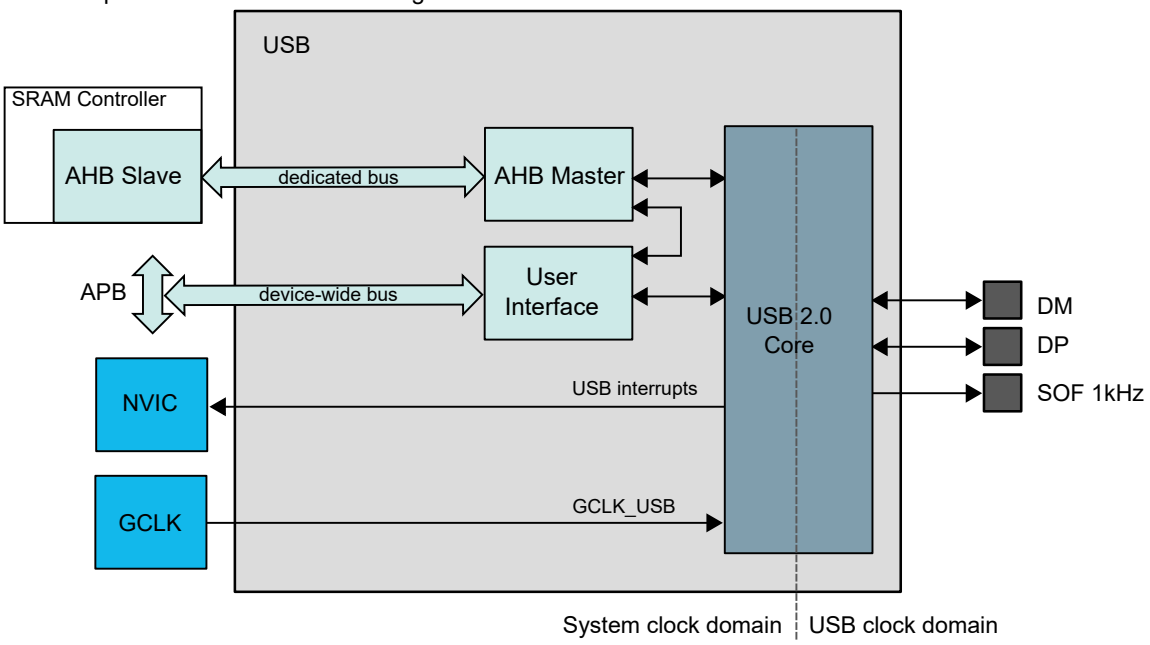
13.25.2 Features

- Compatible with the USB 2.1 specification
- USB Embedded Host and Device mode
- Supports full (12Mbit/s) and low (1.5Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1kHz SOF clock available on external pin
- Device mode
  - Supports 8 IN endpoints and 8 OUT endpoints
  - No endpoint size limitations
  - Built-in DMA with multi-packet and dual bank for all endpoints
  - Supports feedback endpoint
  - Supports crystal less clock
- Host mode
  - Supports 8 physical pipes
  - No pipe size limitations
  - Supports multiplexed virtual pipe on one physical pipe to allow an unlimited USB tree
  - Built-in DMA with multi-packet support and dual bank for all pipes
  - Supports feedback endpoint
  - Supports the USB 2.0 Phase-locked SOFs feature

13.25.3 USB Block Diagram

Figure 13-176. High-speed Implementation: USB Block Diagram

LS/FS Implementation: USB Block Diagram



13.25.4 Signal Description

Pin Name	Pin Description	Type
DM	Data -: Differential Data Line - Port	Input/Output
DP	Data +: Differential Data Line + Port	Input/Output
SOF 1kHZ	SOF Output	Output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.25.5 Product Dependencies

In order to use this peripheral module, other parts of the system must be configured correctly, as described below.

13.25.5.1 I/O Lines

The USB pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the USB pins to their peripheral functions.

A 1kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.

13.25.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.25.5.3 Clocks

The USB bus clock (CLK_USB_AHB) can be enabled and disabled in the Power Manager, and the default state of CLK_USB_AHB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_USB) is required to clock the USB. This clock must be configured and enabled in the Generic Clock Controller before using the USB. Refer to *GCLK - Generic Clock Controller* for further details.

This generic clock is asynchronous to the bus clock (CLK_USB_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *GCLK Synchronization* for further details.

The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation. To follow the USB data rate at 12Mbit/s in full-speed mode, the CLK_USB_AHB clock should be at minimum 8MHz.

Clock recovery is achieved by a digital phase-locked loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, refer to *USB Clock Recovery Module*.

Related Links

13.5 GCLK - Generic Clock Controller

13.5.6.6 Synchronization

USB Clock Recovery Module

13.6.6.2.6 Peripheral Clock Masking

13.25.5.4 DMA

The USB has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

13.25.5.5 Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.25.5.6 Events

Not applicable.

13.25.5.7 Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.25.5.8 Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Device Interrupt Flag (INTFLAG) register
- Endpoint Interrupt Flag (EPINTFLAG) register
- Host Interrupt Flag (INTFLAG) register
- Pipe Interrupt Flag (PINTFLAG) register

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

13.25.5.9 Analog Connections

Not applicable.

13.25.5.10 Calibration

The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.

For details on Pad Calibration, refer to Pad Calibration (13.25.8.1.6 PADCAL) register.

Related Links

10.4 NVM Software Calibration Area Mapping

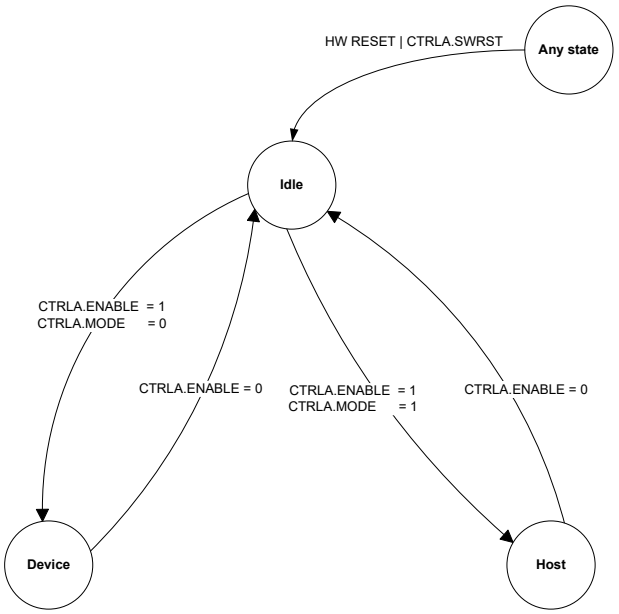
13.25.6 Functional Description

13.25.6.1 USB General Operation

13.25.6.1.1 Initialization

After a hardware reset, the USB is disabled. The user should first enable the USB (CTRLA.ENABLE) in either device mode or host mode (CTRLA.MODE).

Figure 13-177. General States



After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption.
- The USB pad is in suspend mode.
- The internal states and registers of the device and host are reset.

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the CTRLA register for details.

The user can configure pads and speed before enabling the USB by writing to the Operating Mode bit in the Control A register (CTRLA.MODE) and the Speed Configuration field in the Control B register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode or host mode (according to CTRLA.MODE).

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.

Refer to 13.25.6.2 USB Device Operations for the basic operation of the device mode.

Refer to 13.25.6.3 Host Operations for the basic operation of the host mode.

Related Links

10.4 NVM Software Calibration Area Mapping

13.25.6.2 USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

13.25.6.2.1 Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

13.25.6.2.2 Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB master (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to the Endpoint Descriptor structure in 13.25.8.4.1 Endpoint Descriptor Structure.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

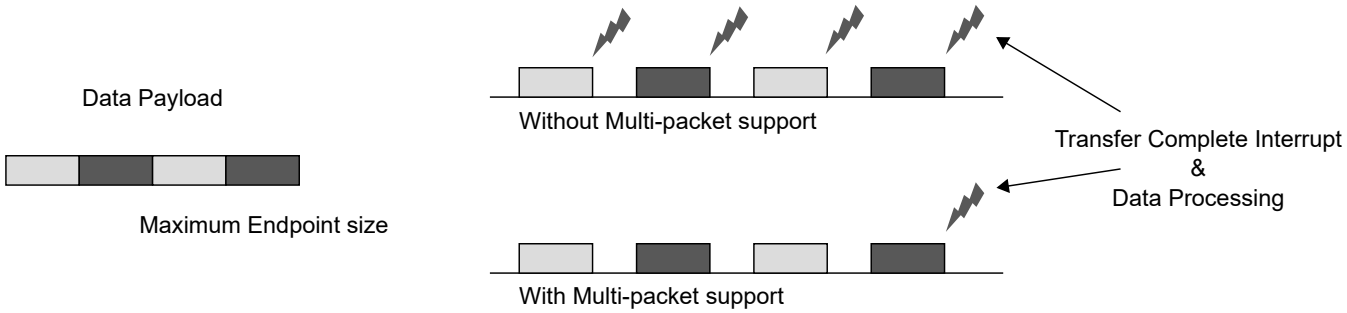
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

13.25.6.2.3 Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

Figure 13-178. Multi-Packet Feature - Reduction of CPU Overhead



13.25.6.2.4 USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

13.25.6.2.5 Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

13.25.6.2.6 Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

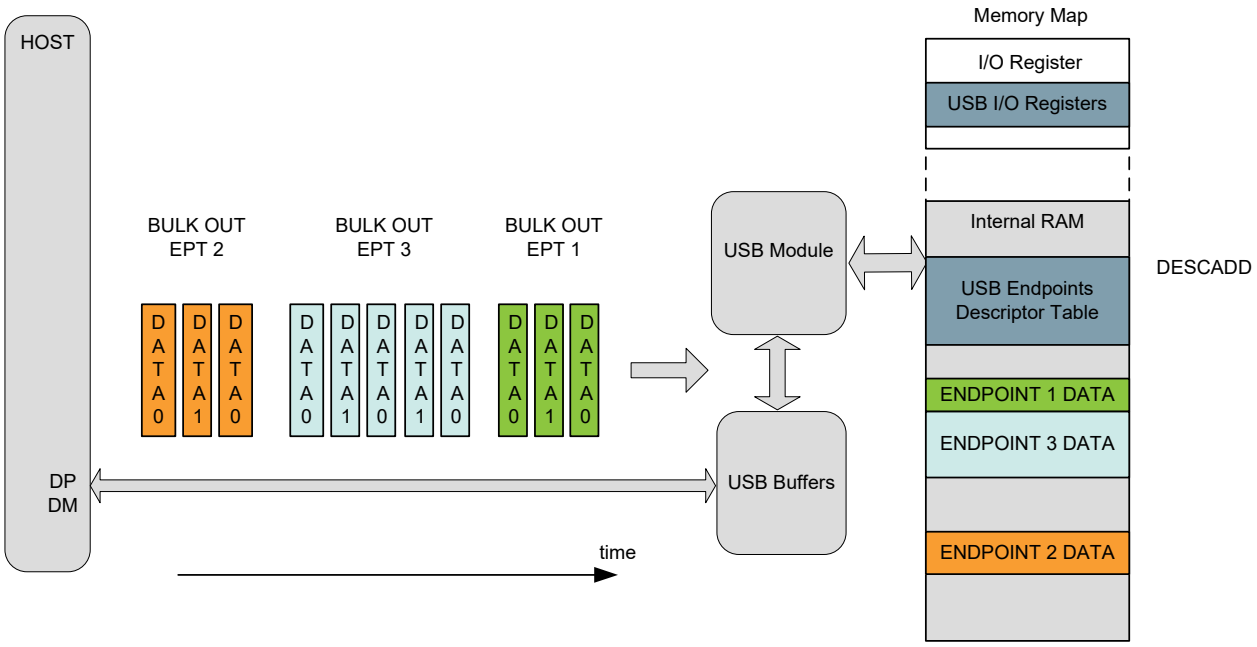
When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

13.25.6.2.7 Management of OUT Transactions

Figure 13-179. OUT Transfer: Data Packet Host to USB Device



When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BK0RDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

13.25.6.2.8 Multi-Packet Transfers for OUT Endpoint

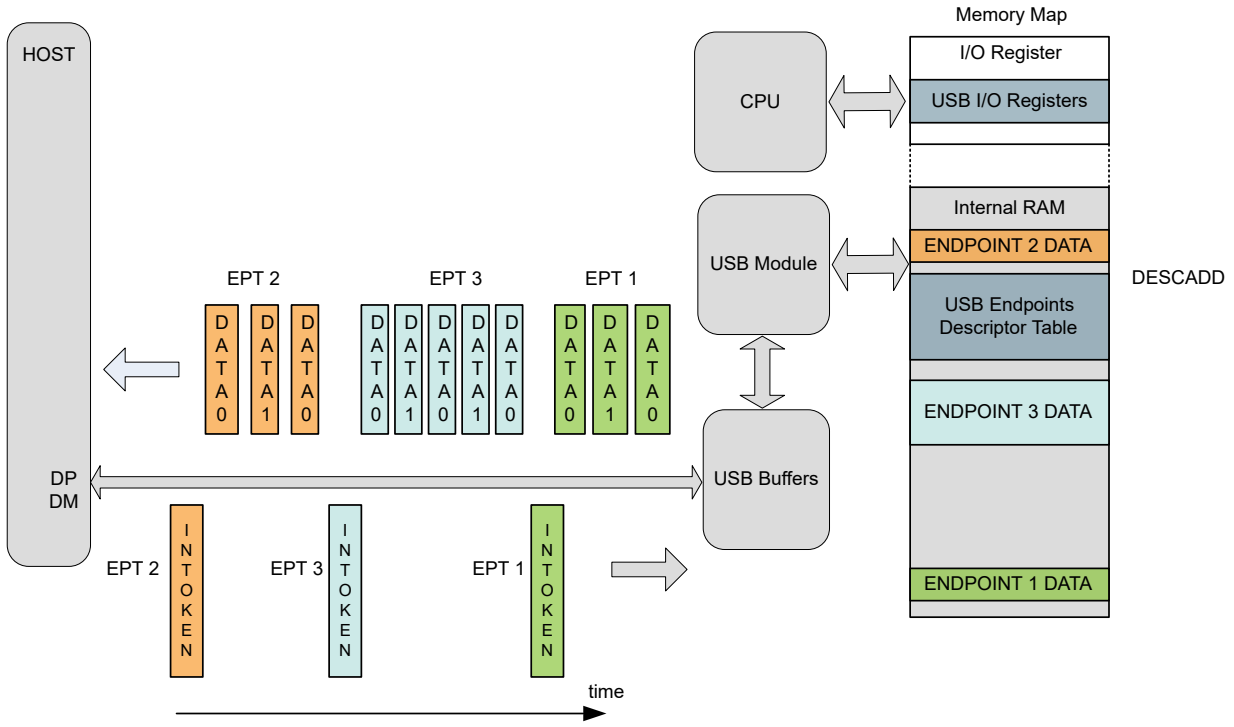
The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.

13.25.6.2.9 Management of IN Transactions

Figure 13-180. IN Transfer: Data Packet USB Device to Host After Request from Host



When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

13.25.6.2.10 Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

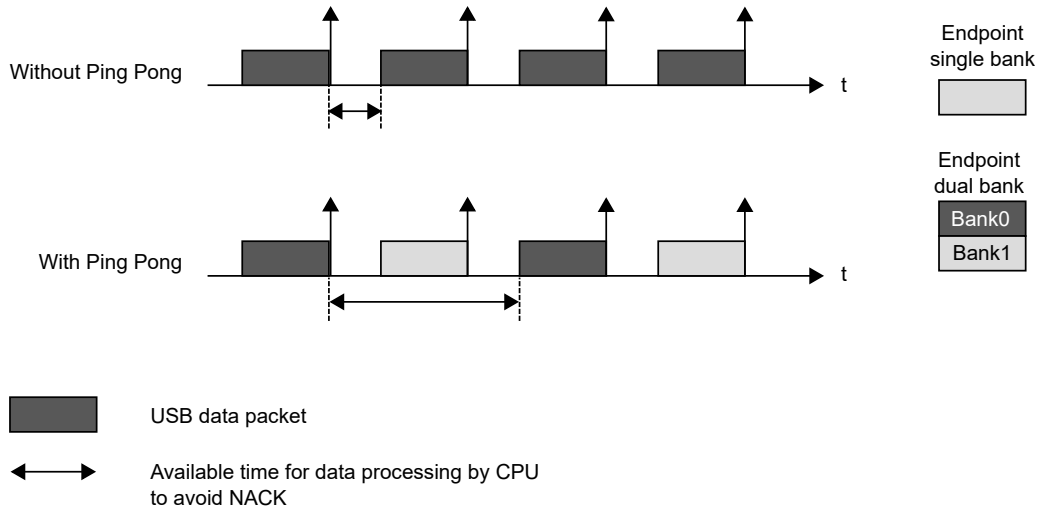
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

13.25.6.2.11 Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

Figure 13-181. Ping-Pong Overview



The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAIL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAIL1 in EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

13.25.6.2.12 Feedback Operation

Feedback endpoints are endpoints with same the address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service. The feedback endpoint always has the opposite direction from the data endpoint.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

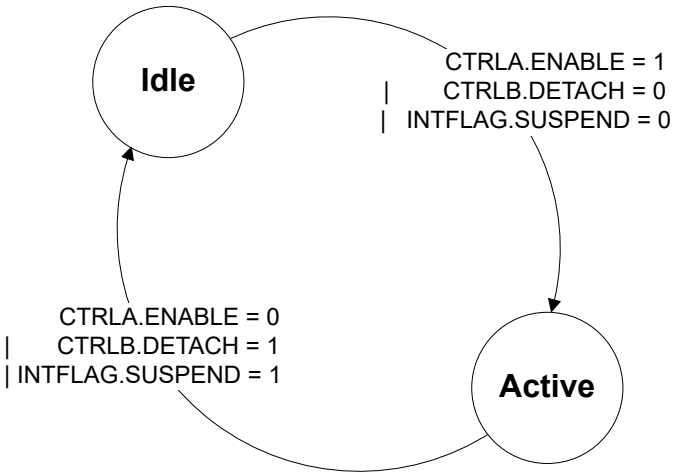
Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0= Isochronous OUT, PCKSIZE.SIZE = 512.

13.25.6.2.13 Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in device mode.

Figure 13-182. Pad Behavior

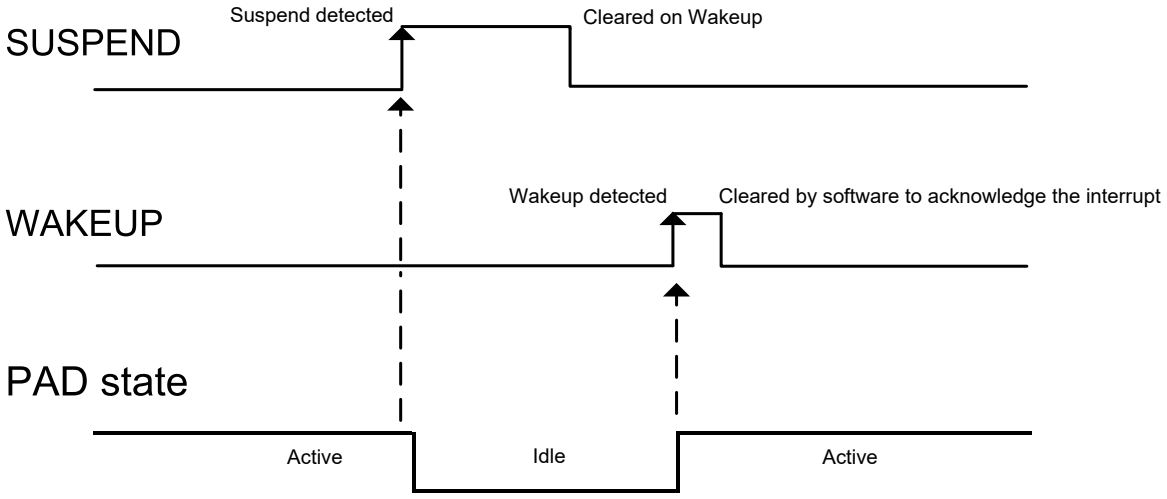


In Idle state, the pad is in low power consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

Figure 13-183. Pad Events



The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit in INTFLAG(INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

13.25.6.2.14 Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit in CTRLB(CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit in INTFLAG(INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

13.25.6.2.15 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint n and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit in INTFLAG(INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in 13.25.8.4.4 EXTREG register).

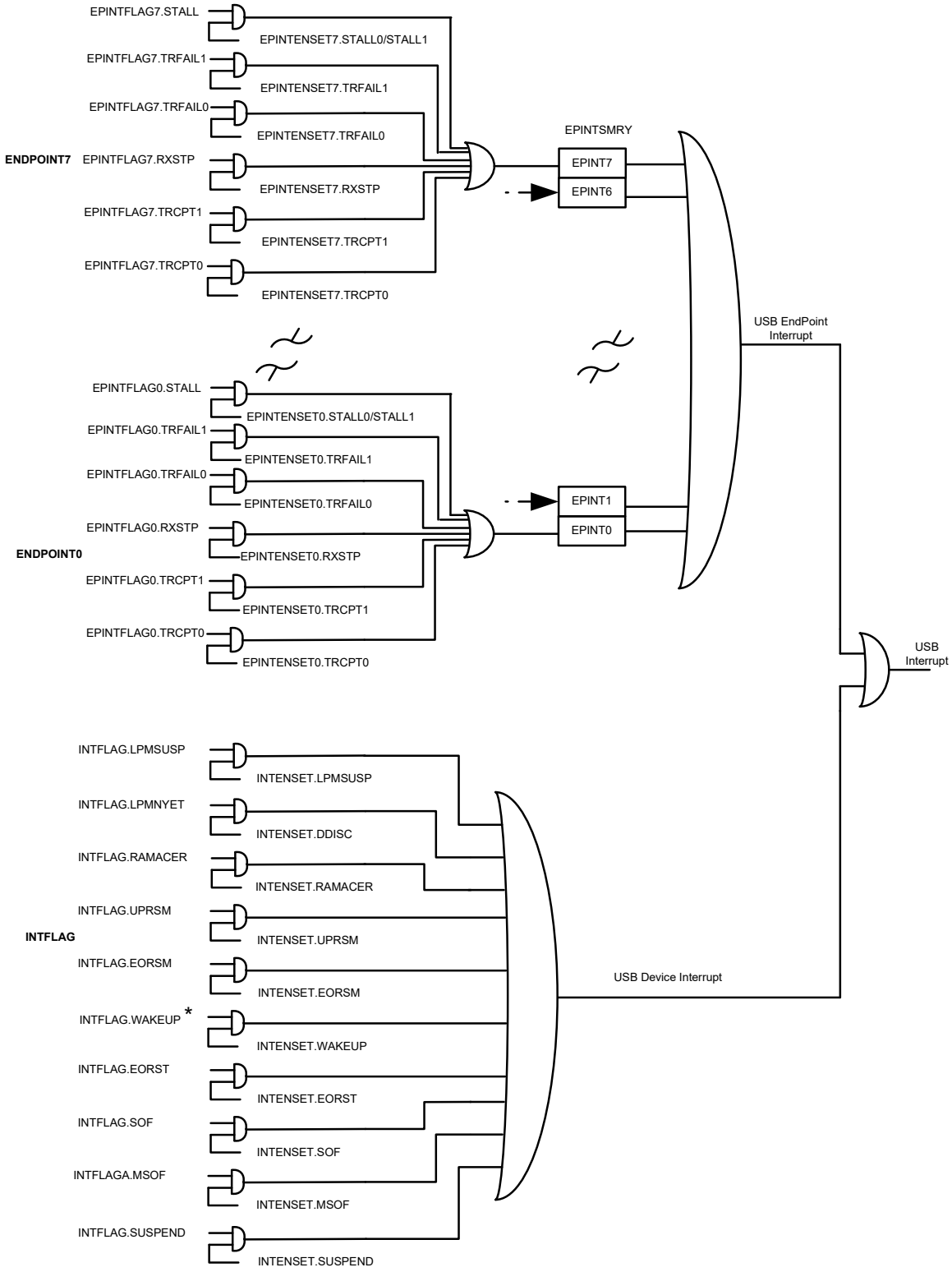
Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section 13.25.6.2.13 Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag INTFLAG(INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit in INTENCLR/SET (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with a STALL or no handshake, no flag is set, and the transaction is ignored.

13.25.6.2.16 USB Device Interrupt

Figure 13-184. Device Interrupt



* Asynchronous interrupt

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

13.25.6.3 Host Operations

This section gives an overview of the USB module Host operation during normal transactions. For more details on general USB and USB protocol, refer to Universal Serial Bus Specification revision 2.1.

13.25.6.3.1 Device Detection and Disconnection

Prior to device detection the software must set the VBUS is OK bit in CTRLB (CTRLB.VBUSOK) register when the VBUS is available. This notifies the USB host that USB operations can be started. When the bit CTRLB.VBUSOK is zero and even if the USB HOST is configured and enabled, host operation is halted. Setting the bit CTRLB.VBUSOK will allow host operation when the USB is configured.

The Device detection is managed by the software using the Line State field in the Host Status (STATUS.LINESTATE) register. The device connection is detected by the host controller when DP or DM is pulled high, depending of the speed of the device.

The device disconnection is detected by the host controller when both DP and DM are pulled down using the STATUS.LINESTATE registers.

The Device Connection Interrupt bit in INTFLAG (INTFLAG.DCONN) is set if a device connection is detected.

The Device Disconnection Interrupt bit in INTFLAG (INTFLAG.DDISC) is set if a device disconnection is detected.

13.25.6.3.2 Host Terminology

In host mode, the term pipe is used instead of endpoint. A host pipe corresponds to a device endpoint, refer to "Universal Serial Bus Specification revision 2.1." for more information.

13.25.6.3.3 USB Reset

The USB sends a USB reset signal when the user writes a one to the USB Reset bit in CTRLB (CTRLB.BUSRESET). When the USB reset has been sent, the USB Reset Sent Interrupt bit in the INTFLAG (INTFLAG.RST) is set and all pipes will be disabled.

If the bus was previously in a suspended state (Start of Frame Generation Enable bit in CTRLB (CTRLB.SOFE) is zero) the USB will switch it to the Resume state, causing the bus to asynchronously set the Host Wakeup Interrupt flag (INTFLAG.WAKEUP). The CTRLB.SOFE bit will be set in order to generate SOFs immediately after the USB reset.

During USB reset the following registers are cleared:

- All Host Pipe Configuration register (PCFG)
- Host Frame Number register (FNUM)
- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Host Start-of-Frame Control register (HSOFC)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

After the reset the user should check the Speed Status field in the Status register (STATUS.SPEED) to find out the current speed according to the capability of the peripheral.

13.25.6.3.4 Pipe Configuration

Pipe data can be placed anywhere in the RAM. The USB controller accesses these pipes directly through the AHB master (built-in DMA) with the help of the pipe descriptors. The base address of the pipe descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to 13.25.8.7.1 Pipe Descriptor Structure.

Before using a pipe, the user should configure the direction and type of the pipe in Type of Pipe field in the Host Pipe Configuration register (PCFG.PTYPE). The pipe descriptor registers should be initialized to known values before using the pipe, so that the USB controller does not read the random values from the RAM.

The Pipe Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported by the device for the endpoint associated with this pipe. The Address of Data Buffer register (ADDR) should be set to the data buffer used for pipe transfers.

The Pipe Bank bit in PCFG (PCFG.BK) should be set to one if dual banking is desired. Dual bank is not supported for Control pipes.

The Ram Access Interrupt bit in Host Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during an OUT stage.

When a pipe is disabled, the following registers are cleared for that pipe:

- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

13.25.6.3.5 Pipe Activation

A disabled pipe is inactive, and will be reset along with its context registers (pipe registers for the pipe n). Pipes are enabled by writing Type of the Pipe in PCFG (PCFG.PTYPE) to a value different than 0x0 (disabled).

When a pipe is enabled, the Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE) is set. This allow the user to complete the configuration of the pipe, without starting a USB transfer.

When starting an enumeration, the user retrieves the device descriptor by sending an GET_DESCRIPTOR USB request. This descriptor contains the maximal packet size of the device default control endpoint (bMaxPacketSize0) which the user should use to reconfigure the size of the default control pipe.

13.25.6.3.6 Pipe Address Setup

Once the device has answered the first host requests with the default device address 0, the host assigns a new address to the device. The host controller has to send a USB reset to the device and a SET_ADDRESS(addr) SETUP request with the new address to be used by the device. Once this SETUP transaction is complete, the user writes the new address to the Pipe Device Address field in the Host Control Pipe register (CTRL_PIPE.PDADDR) in Pipe descriptor. All following requests by this pipe will be performed using this new address.

13.25.6.3.7 Suspend and Wakeup

Setting CTRLB.SOFE to zero when in host mode will cause the USB to cease sending Start-of-Frames on the USB bus and enter the Suspend state. The USB device will enter the Suspend state 3ms later.

Before entering suspend by writing CTRLB.SOFE to zero, the user must freeze the active pipes by setting their PSTATUS.FREEZE bit. Any current on-going pipe will complete its transaction, and then all pipes will be inactive. The user should wait at least 1 complete frame before entering the suspend mode to avoid any data loss.

The device can awaken the host by sending an Upstream Resume (Remote Wakeup feature). When the host detects a non-idle state on the USB bus, it sets the INTFLAG.WAKEUP. If the non-idle bus state corresponds to an Upstream Resume (K state), the Upstream Resume Received Interrupt bit in INTFLAG (INTFLAG.UPRSM) is set and the user must generate a Downstream Resume within 1 ms and for at least 20 ms. It is required to first write a one to the Send USB Resume bit in CTRLB (CTRLB.RESUME) to respond to the upstream resume with a downstream resume. Alternatively, the host can resume from a suspend state by sending a Downstream Resume on the USB bus (CTRLB.RESUME set to 1). In both cases, when the downstream resume is completed, the CTRLB.SOFE bit is automatically set and the host enters again the active state.

13.25.6.3.8 Phase-locked SOFs

To support the Synchronous Endpoints capability, the period of the emitted Start-of-Frame is maintained while the USB connection is not in the active state. This does not apply for the disconnected/connected/reset states. It applies for active/idle/suspend/resume states. The period of Start-of-Frame will be 1ms when the USB connection is in active state and an integer number of milli-seconds across idle/suspend/resume states.

To ensure the Synchronous Endpoints capability, the GCLK_USB clock must be kept running. If the GCLK_USB is interrupted, the period of the emitted Start-of-Frame will be erratic.

13.25.6.3.9 Management of Control Pipes

A control transaction is composed of three stages:

- SETUP
- Data (IN or OUT)
- Status (IN or OUT)

The user has to change the pipe token according to each stage using the Pipe Token field in PCFG (PCFG.PTOKEN).

For control pipes only, the token is assigned a specific initial data toggle sequence:

- SETUP: Data0
- IN: Data1
- OUT: Data1

13.25.6.3.10 Management of IN Pipes

IN packets are sent by the USB device controller upon IN request reception from the host. All the received data from the device to the host will be stored in the bank provided the bank is empty. The pipe and its descriptor in RAM must be configured.

The host indicates it is able to receive data from the device by clearing the Bank 0/1 Ready bit in PSTATUS (PSTATUS.BK0/1RDY), which means that the memory for the bank is available for new USB transfer.

The USB will perform IN requests as long as the pipe is not frozen by the user.

The generation of IN requests starts when the pipe is unfrozen (PSTATUS.PFREEZE is set to zero).

When the current bank is full, the Transmit Complete 0/1 bit in PINTFLAG (PINTFLAG.TRCPT0/1) will be set and trigger an interrupt if enabled and the PSTATUS.BK0/1RDY bit will be set.

PINTFLAG.TRCPT0/1 must be cleared by software to acknowledge the interrupt. This is done by writing a one to the PINTFLAG.TRCPT0/1 of the addressed pipe.

The user reads the PCKSIZE.BYTE_COUNT to know how many bytes should be read.

To free the bank the user must read the IN data from the address ADDR in the pipe descriptor and clear the PKSTATUS.BK0/1RDY bit. When the IN pipe is composed of multiple banks, a successful IN transaction will switch to the next bank. Another IN request will be performed by the host as long as the PSTATUS.BK0/1RDY bit for that bank is set. The PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1RDY will be updated accordingly.

The user can follow the current bank looking at Current Bank bit in PSTATUS (PSTATUS.CURBK) and by looking at Data Toggle for IN pipe bit in PSTATUS (PSTATUS.DTGLIN).

When the pipe is configured as single bank (Pipe Bank bit in PCFG (PCFG.BK) is 0), only PINTFLAG.TRCPT0 and PSTATUS.BK0 are used. When the pipe is configured as dual bank (PCFG.BK is 1), both PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1 are used.

13.25.6.3.11 Management of OUT Pipes

OUT packets are sent by the host. All the data stored in the bank will be sent to the device provided the bank is filled. The pipe and its descriptor in RAM must be configured.

The host can send data to the device by writing to the data bank 0 in single bank or the data bank 0/1 in dual bank.

The generation of OUT packet starts when the pipe is unfrozen (PSTATUS.PFREEZE is zero).

The user writes the OUT data to the data buffer pointer by ADDR in the pipe descriptor and allows the USB to send the data by writing a one to the PSTATUS.BK0/1RDY. This will also cause a switch to the next bank if the OUT pipe is part of a dual bank configuration.

PINTFLAGn.TRCPT0/1 must be cleared before setting PSTATUS.BK0/1RDY to avoid missing an PINTFLAGn.TRCPT0/1 event.

13.25.6.3.12 Alternate Pipe

The user has the possibility to run sequentially several logical pipes on the same physical pipe. It allows addressing of any device endpoint of any attached device on the bus.

Before switching pipe, the user should save the pipe context (Pipe registers and descriptor for pipe n).

After switching pipe, the user should restore the pipe context (Pipe registers and descriptor for pipe n) and in particular PCFG, and PSTATUS.

13.25.6.3.13 Data Flow Error

This error exists only for isochronous and interrupt pipes for both IN and OUT directions. It sets the Transmit Fail bit in PINTFLAG (PINTFLAG.TRFAIL), which triggers an interrupt if the Transmit Fail bit in PINTENCLR/SET(PINTENCLR/SET.TRFAIL) is set. The user must check the Pipe Interrupt Summary register (PINTSMRY) to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the Pipe Bank Status register (STATUS_BK) for each bank. If the Error Flow bit in the STATUS_BK

(STATUS_BK.ERRORFLOW) is set then the user is able to determine the origin of the data flow error. As the user knows that the endpoint is an IN or OUT the error flow can be deduced as OUT underflow or as an IN overflow.

An underflow can occur during an OUT stage if the host attempts to send data from an empty bank. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

An overflow can occur during an IN stage if the device tries to send a packet while the bank is full. Typically this occurs when a CPU is not fast enough. The packet data is not written to the bank and is lost. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

13.25.6.3.14 CRC Error

This error exists only for isochronous IN pipes. It sets the PINTFLAG.TRFAIL, which triggers an interrupt if PINTENCLR/SET.TRFAIL is set. The user must check the PINTSMRY to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the bank descriptor STATUS_BK for each bank and if the CRC Error bit in STATUS_BK (STATUS_BK.CRCERR) is set then the user is able to determine the origin of the CRC error. A CRC error can occur during the IN stage if the USB detects a corrupted packet. The IN packet will remain stored in the bank and PINTFLAG.TRCPT0/1 will be set.

13.25.6.3.15 PERR Error

This error exists for all pipes. It sets the PINTFLAG.PERR Interrupt, which triggers an interrupt if PINTFLAG.PERR is set. The user must check the PINTSMRY register to find the pipe that can cause an interrupt.

A PERR error occurs if one of the error fields in the STATUS_PIPE register in the Host pipe descriptor is set and the Error Count field in STATUS_PIPE (STATUS_PIPE.ERCNT) exceeds the maximum allowed number of Pipe error(s) as defined in the Pipe Error Max Number field in CTRL_PIPE (CTRL_PIPE.PERMAX). Refer to 13.25.8.7.7 STATUS_PIPE.

If one of the error fields in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

13.25.6.3.16 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Host.

An EXTENDED LPM transaction can be transmitted by any enabled pipe. The PCFGn.PTYPE should be set to EXTENDED. Other fields as PCFG.PTOKEN, PCFG.BK and PCKSIZE.SIZE are irrelevant in this configuration. The user should also set the EXTREG.VARIABLE in the descriptor as described in 13.25.8.7.4 EXTREG register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

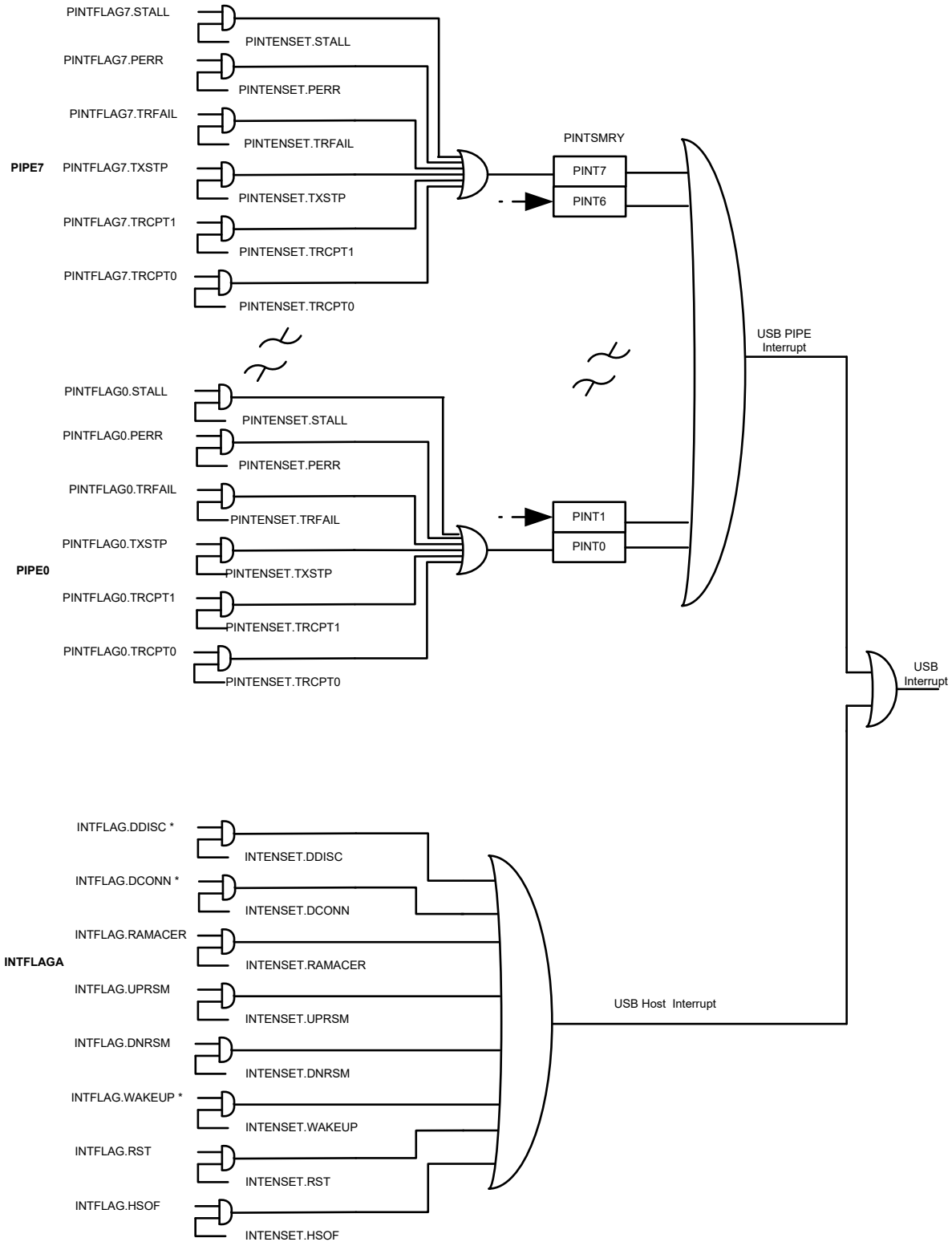
To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. See 13.25.8.7.4 EXTREG.

When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.

13.25.6.3.17 Host Interrupt

Figure 13-185. Host Interrupt



* Asynchronous interrupt

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

13.25.7 Register Summary

The register mapping depends on the Operating Mode field in the Control A register (CTRLA.MODE). The register summary is detailed below.

13.25.7.1 Common Device Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MODE					RUNSTBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x24	DESCADD	7:0	DESCADD[7:0]							
0x25		15:8	DESCADD[15:8]							
0x26		23:16	DESCADD[23:16]							
0x27		31:24	DESCADD[31:24]							
0x28	PADCAL	7:0	TRANSN[1:0]			TRANSP[4:0]				
0x29		15:8		TRIM[2:0]				TRANSN[4:2]		

13.25.7.2 Device Summary

Table 13-79. General Device Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x08	CTRLB	7:0				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
0x09		15:8					LPMHDSK[1:0]		GNAK	
0x0A	DADD		ADDEN	DADD[6:0]						
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8	FNCERR		FNUM[10:5]					
0x12	Reserved									
0x14	INTENCLR	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x15		15:8							LPMSUSP	LPMNYET
0x16	Reserved									
0x17	Reserved									
0x18	INTENSET	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x19		15:8							LPMSUSP	LPMNYET
0x1A	Reserved									
0x1B	Reserved									
0x1C	INTFLAG	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x1D		15:8							LPMSUSP	LPMNYET
0x1E	Reserved									
0x1F	Reserved									
0x20	13.25.8.2.8	7:0	EPINT[7:0]							
0x21	EPINTSMRY	15:8	EPINT[15:8]							
0x22	Reserved									
0x23	Reserved									

Table 13-80. Device Endpoint Register n

Offset	Name	Bit Pos.								
0x1m0	EPCFGn	7:0		EPTYPE1[1:0]				EPTYPE0[1:0]		

.....continued										
Offset	Name	Bit Pos.								
0x1m1	Reserved									
0x1m2	Reserved									
0x1m3	Reserved									
0x1m4	EPSTATUSCLRn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m5	EPSTATUSSETn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m6	13.25.8.3.4 EPSTATUSn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m7	13.25.8.3.5 EPINTFLAGn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m8	EPINTENCLRn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m9	EPINTENSETn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1mA	Reserved									
0x1mB	Reserved									

Table 13-81. Device Endpoint n Descriptor Bank 0

Offset 0x n0 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]			BYTE_COUNT[13:8]				
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	EXTREG	7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x09		15:8		VARIABLE[10:4]						
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

Table 13-82. Device Endpoint n Descriptor Bank 1

Offset 0x n0 + 0x10 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]			BYTE_COUNT[13:8]				
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	Reserved	7:0								
0x09	Reserved	15:8								
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

13.25.7.3 Host Summary

Table 13-83. General Host Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x08	CTRLB	7:0		TSTK	TSTJ		SPDCONF[1:0]		RESUME	
0x09		15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
0x0A	HSOFC	7:0	FLENCE				FLENC[3:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8			FNUM[10:5]					
0x12	FLENHIGH	7:0	FLENHIGH[7:0]							
0x14	INTENCLR	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x15		15:8							DDISC	DCONN
0x16	Reserved									
0x17	Reserved									
0x18	INTENSET	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x19		15:8							DDISC	DCONN
0x1A	Reserved									
0x1B	Reserved									
0x1C	INTFLAG	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1D		15:8							DDISC	DCONN
0x1E	Reserved									
0x1F	Reserved									
0x20	PINTSMRY	7:0	PINT[7:0]							
0x21		15:8	PINT[15:8]							
0x22	Reserved									
0x23										

Table 13-84. Host Pipe Register n

Offset	Name	Bit Pos.								
0x1m0	PCFGn	7:0				PTYPE[2:0]		BK	PTOKEN[1:0]	
0x1m1	Reserved									
0x1m2	Reserved									
0x1m3	BINTERVAL	7:0	BINTERVAL[7:0]							
0x1m4	PSTATUSCLRn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m5	PSTATUSETn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m6	PSTATUSn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m7	PINTFLAGn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1m8	PINTENCLRn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1m9	PINTENSETn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1mA	Reserved									
0x1mB	Reserved									

Table 13-85. Host Pipe n Descriptor Bank 0

Offset 0x n0 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]			BYTE_COUNT[13:8]				
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	EXTREG	7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x09		15:8		VARIABLE[10:4]						
0x0A	13.25.8.7.5 STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B		15:8								
0x0C	CTRL_PIPE	7:0		PDADDR[6:0]						
0x0D		15:8	PEPMAX[3:0]				PEPNUM[3:0]			
0x0E	13.25.8.7.7 STATUS_PIPE	7:0	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
0x0F		15:8								

Table 13-86. Host Pipe n Descriptor Bank 1

Offset 0x n0 +0x10 +index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0			BYTE_COUNT[13:8]				
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08		7:0								
0x09		15:8								
0x0A	13.25.8.7.5 STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B		15:8								
0x0C		7:0								
0x0D		15:8								
0x0E	13.25.8.7.7 STATUS_PIPE	7:0	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
0x0F		15:8								

13.25.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to the 13.25.5.8 Register Access Protection, *PAC - Peripheral Access Controller* and *GCLK Synchronization* for details.

13.25.8.1 Communication Device Host Registers

13.25.8.1.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronised

Bit	7	6	5	4	3	2	1	0
	MODE					RUNSDTBY	ENABLE	SWRST
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – MODE Operating Mode
This bit defines the operating mode of the USB.

Value	Description
0	USB Device mode
1	USB Host mode

Bit 2 – RUNSDTBY Run in Standby Mode
This bit is Enable-Protected.

Value	Description
0	USB clock is stopped in standby mode.
1	USB clock is running in standby mode

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Synchronization status enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is Write-Synchronized.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is Write-Synchronized.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.25.8.1.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x02
Reset: 0x0000
Property: -

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started.

13.25.8.1.3 QOS Control

Name: QOSCTRL
Offset: 0x03
Reset: 0x0F
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					DQOS[1:0]		CQOS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 3:2 – DQOS[1:0] Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to *SRAM Quality of Service*.

Bits 1:0 – CQOS[1:0] Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to *SRAM Quality of Service*.

Related Links
11.4.3 SRAM Quality of Service

13.25.8.1.4 Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D
Reset: 0xXXXX
Property: Read only

Bit	7	6	5	4	3	2	1	0
		FSMSTATE[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	1

Bits 6:0 – FSMSTATE[6:0] Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

Value	Name	Description
0x01	OFF (L3)	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON (L0)	Corresponds to the Idle and Active states.
0x04	SUSPEND (L2)	
0x08	SLEEP (L1)	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

13.25.8.1.5 Descriptor Address

Name: DESCADD
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DESCADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DESCADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DESCADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DESCADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DESCADD[31:0] Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.

13.25.8.1.6 Pad Calibration

Name: PADCAL
Offset: 0x28
Reset: 0x0000
Property: PAC Write-Protection

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy.
Refer to *NVM Software Calibration Area Mapping* for further details.

Refer to for further details.

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	TRANSN[1:0]			TRANSP[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0] Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0] Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0] Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

Related Links
10.4 NVM Software Calibration Area Mapping

13.25.8.2 Device Registers - Common

13.25.8.2.1 Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0001
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access				R	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 11:10 – LPMHDSK[1:0] Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

Value	Description
0x0	No handshake. LPM is not supported.
0x1	ACK
0x2	NYET
0x3	Reserved

Bit 9 – GNAK Global NAK
This bit configures the operating mode of the NAK.
This bit is not synchronized.

Value	Description
0	The handshake packet reports the status of the USB transaction
1	A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status

Bit 4 – NREPLY No reply excepted SETUP Token
This bit is cleared by hardware when receiving a SETUP packet.
This bit has no effect for any other endpoint but endpoint 0.

Value	Description
0	Disable the “NO_REPLY” feature: Any transaction to endpoint 0 will be handled according to the USB2.0 standard.
1	Enable the “NO_REPLY” feature: Any transaction to endpoint 0 will be ignored except SETUP.

Bits 3:2 – SPDCONF[1:0] Speed Configuration
These bits select the speed configuration.

Value	Description
0x0	FS: Full-speed
0x1	LS: Low-speed
0x2	Reserved
0x3	Reserved

Bit 1 – UPRSM Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.

Value	Description
0	Writing a zero to this bit has no effect.
1	Writing a one to this bit will generate an upstream resume to the host for a remote wakeup.

Bit 0 – DETACH Detach

Value	Description
0	The device is attached to the USB bus so that communications may occur.
1	It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus.

13.25.8.2.2 Device Address

Name: DADD
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ADDEN	DADD[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ADDEN Device Address Enable
This bit is cleared when a USB reset is received.

Value	Description
0	Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0.
1	Writing a one will activate the DADD field (USB device address).

Bits 6:0 – DADD[6:0] Device Address
These bits define the device address. The DADD register is reset when a USB reset is received.

13.25.8.2.3 Status

Name: STATUS
Offset: 0x0C
Reset: 0x40
Property: -

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R/W	R/W		
Reset	0	1			0	1		

Bits 7:6 – LINESTATE[1:0] USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0] Speed Status
These bits define the current speed used of the device
.

SPEED[1:0]	SPEED STATUS
0x0	Low-speed mode
0x1	Full-speed mode
0x2	Reserved
0x3	Reserved

13.25.8.2.4 Device Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: Read only

Bit	15	14	13	12	11	10	9	8
	FNCERR		FNUM[10:5]					
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FNCERR Frame Number CRC Error
This bit is cleared upon receiving a USB reset.
This bit is set when a corrupted frame number (or micro-frame number) is received.
This bit and the SOF (or MSOF) interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0] Frame Number
These bits are cleared upon receiving a USB reset.
These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

Bits 2:0 – MFNUM[2:0] Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).
These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.

13.25.8.2.5 Device Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – EORSM End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set.

Bit 4 – WAKEUP Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – EORST End of Reset Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set.

Bit 2 – SOF Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set.

Bit 0 – SUSPEND Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set.

13.25.8.2.6 Device Interrupt Enable Set

Name: INTENSET
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – EORSM End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST End of Reset Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 0 – SUSPEND Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled.

13.25.8.2.7 Device Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENCLR/SET.LPMSUSP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMSUSP Interrupt Flag.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENCLR/SET.LPMNYET is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMNYET Interrupt Flag.

Bit 7 – RAMACER RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENCLR/SET.RAMACER is one.
Writing a zero to this bit has no effect.

Bit 6 – UPRSM Upstream Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENCLR/SET.UPRSM is one.
Writing a zero to this bit has no effect.

Bit 5 – EORSM End Of Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENCLR/SET.EORSM is one.
Writing a zero to this bit has no effect.

Bit 4 – WAKEUP Wake Up Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENCLR/SET.WAKEUP is one.
Writing a zero to this bit has no effect.

Bit 3 – EORST End of Reset Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB “End of Reset” has been detected and will generate an interrupt if INTENCLR/SET.EORST is one.
Writing a zero to this bit has no effect.

Bit 2 – SOF Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB “Start-of-Frame” has been detected (every 1 ms) and will generate an interrupt if INTENCLR/SET.SOF is one.
The FNUM is updated. In High Speed mode, the MFNUM register is cleared.
Writing a zero to this bit has no effect.

Bit 0 – SUSPEND Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB “Suspend” idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENCLR/SET.SUSPEND is one.
Writing a zero to this bit has no effect.

13.25.8.2.8 Endpoint Interrupt Summary

Name: EPINTSMRY
Offset: 0x20
Reset: 0x00000000
Property: —

Bit	15	14	13	12	11	10	9	8
	EPINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EPINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EPINT[15:0] EndPoint Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. See the 13.25.8.3.5 EPINTFLAGn register in the device EndPoint section.
This bit will be cleared when no interrupts are pending for EndPoint n.

13.25.8.3 Device Registers - Endpoint

13.25.8.3.1 Device Endpoint Configuration register n

Name: EPCFGn
Offset: 0x100 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		EPTYPE1[2:0]				EPTYPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 6:4 – EPTYPE1[2:0] Endpoint Type for IN direction
These bits contains the endpoint type for IN direction.
Upon receiving a USB reset EPCFGn.EPTYPE1 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank1 is disabled.
0x1	Bank1 is enabled and configured as Control IN.
0x2	Bank1 is enabled and configured as Isochronous IN.
0x3	Bank1 is enabled and configured as Bulk IN.
0x4	Bank1 is enabled and configured as Interrupt IN.
0x5	Bank1 is enabled and configured as Dual-Bank OUT (Endpoint type is the same as the one defined in EPTYPE0)
0x6-0x7	Reserved

Bits 2:0 – EPTYPE0[2:0] Endpoint Type for OUT direction
These bits contains the endpoint type for OUT direction.
Upon receiving a USB reset EPCFGn.EPTYPE0 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank0 is disabled.
0x1	Bank0 is enabled and configured as Control SETUP / Control OUT.
0x2	Bank0 is enabled and configured as Isochronous OUT.
0x3	Bank0 is enabled and configured as Bulk OUT.
0x4	Bank0 is enabled and configured as Interrupt OUT.
0x5	Bank0 is enabled and configured as Dual Bank IN (Endpoint type is the same as the one defined in EPTYPE1)
0x6-0x7	Reserved

13.25.8.3.2 EndPoint Status Clear n

Name: EPSTATUSCLRn
Offset: 0x104 + (n * 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1 STALL bank 1 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0 STALL bank 0 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.CURBK bit.

Bit 1 – DTGLIN Data Toggle IN Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT Data Toggle OUT Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the EPSTATUS.DTGLOUT bit.

13.25.8.3.3 EndPoint Status Set n

Name: EPSTATUSSETn
Offset: 0x105 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1 STALL Request bank 1 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0 STALL Request bank 0 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.CURBK bit.

Bit 1 – DTGLIN Data Toggle IN Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT Data Toggle OUT Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the EPSTATUS.DTGLOUT bit.

13.25.8.3.4 EndPoint Status n

Name: EPSTATUSn
Offset: 0x106
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R	R	R	R		R	R	R
Reset	0	0	0	2		0	0	0

Bit 7 – BK1RDY Bank 1 is ready
For Control/OUT direction Endpoints, the bank is empty.
Writing a '1' to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a '1' to the bit EPSTATUSSET.BK1RDY will set this bit.

Value	Description
0	The bank number 1 is not ready: for IN direction Endpoints, the bank is not yet filled in.
1	The bank number 1 is ready: for IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 6 – BK0RDY Bank 0 is ready
Writing a '1' to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a '1' to the bit EPSTATUSSET.BK0RDY will set this bit.

Value	Description
0	The bank number 0 is not ready: for IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 0 is ready: for IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bits 4, 5 – STALLRQ STALL bank x request
Writing a '0' to the bit EPSTATUSCLR.STALLRQ will clear this bit.
Writing a '1' to the bit EPSTATUSSET.STALLRQ will set this bit.
This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQx feature.
1	Enable STALLRQx feature: a STALL handshake will be sent to the host in regards to bank x.

Bit 2 – CURBK Current Bank
Writing a '0' to the bit EPSTATUSCLR.CURBK will clear this bit.
Writing a '1' to the bit EPSTATUSSET.CURBK will set this bit.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 1 – DTGLIN Data Toggle IN Sequence
Writing a '0' to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.
Writing a '1' to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be 0: data 0.
1	The PID of the next expected IN transaction will be 1: data 1.

Bit 0 – DTGLOUT Data Toggle OUT Sequence
Writing a '0' to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.
Writing a '1' to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be 0: data 0.
1	The PID of the next expected OUR transaction will be 1: data 1.

13.25.8.3.5 Device EndPoint Interrupt Flag n

Name: EPINTFLAGn
Offset: 0x107
Reset: 0x00
Property: —

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	2	0	0	2	0	2

Bits 5, 6 – STALL Transmit Stall x Interrupt Flag
This flag is cleared by writing a '1' to the flag.
This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL is '1'.
EPINTFLAG.STALL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is '0'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the STALL Interrupt Flag.

Bit 4 – RXSTP Received Setup Interrupt Flag
This flag is cleared by writing a '1' to the flag.
This flag is set when a Received Setup occurs and will generate an interrupt if EPINTENCLR/SET.RXSTP is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the RXSTP Interrupt Flag.

Bits 2, 3 – TRFAIL Transfer Fail x Interrupt Flag
This flag is cleared by writing a '1' to the flag.
This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL is '1'.
EPINTFLAG.TRFAIL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is '0'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the TRFAIL Interrupt Flag.

Bits 0, 1 – TRCPT Transfer Complete x Interrupt Flag
This flag is cleared by writing a '1' to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT is '1'.
EPINTFLAG.TRCPT is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is '0'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the TRCPT0 Interrupt Flag.

13.25.8.3.6 Device EndPoint Interrupt Enable n

Name: EPINTENCLRn
Offset: 0x108 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENSET) register.

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1 Transmit STALL 1 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmit Stall 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 1 Interrupt Flag is set.

Bit 5 – STALL0 Transmit STALL 0 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmit Stall 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 0 Interrupt Flag is set.

Bit 4 – RXSTP Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Setup Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled and an interrupt request will be generated when the Received Setup Interrupt Flag is set.

Bit 3 – TRFAIL1 Transfer Fail 1 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail 1 interrupt is disabled.
1	The Transfer Fail 1 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 1 Interrupt Flag is set.

Bit 2 – TRFAIL0 Transfer Fail 0 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail bank 0 interrupt is disabled.
1	The Transfer Fail bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 0 Interrupt Flag is set.

Bit 1 – TRCPT1 Transfer Complete 1 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete 1 interrupt is disabled.
1	The Transfer Complete 1 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 1 Interrupt Flag is set.

Bit 0 – TRCPT0 Transfer Complete 0 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete 0 interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 0 Interrupt Flag is set.

13.25.8.3.7 Device Interrupt EndPoint Set n

Name: EPINTENSETn
Offset: 0x109 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENCLR) register. This register is cleared by USB reset or when EPEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1 Transmit Stall 1 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmit bank 1 Stall interrupt.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled.

Bit 5 – STALL0 Transmit Stall 0 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmit bank 0 Stall interrupt.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled.

Bit 4 – RXSTP Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Received Setup interrupt.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled.

Bit 3 – TRFAIL1 Transfer Fail bank 1 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 2 – TRFAIL0 Transfer Fail bank 0 Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 1 – TRCPT1 Transfer Complete bank 1 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete 0 interrupt.

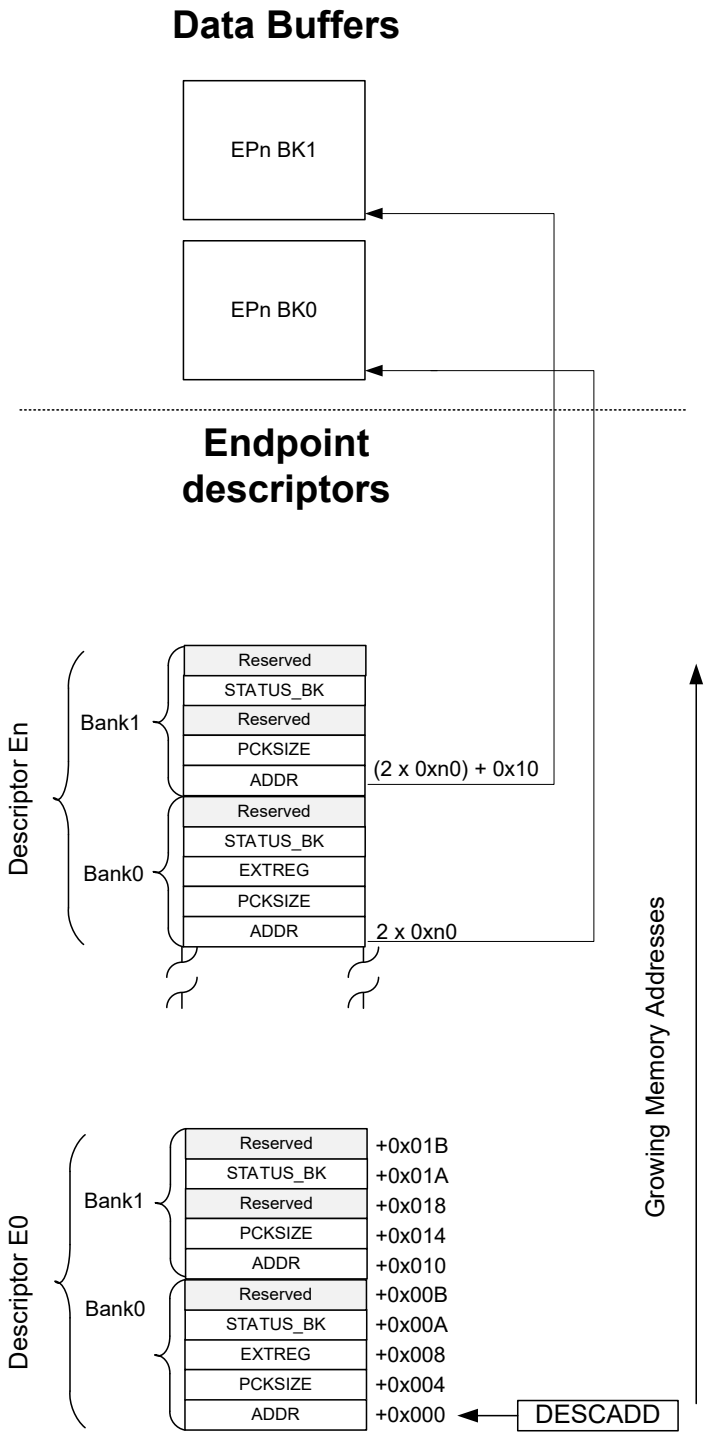
Value	Description
0	The Transfer Complete bank 1 interrupt is disabled.
1	The Transfer Complete bank 1 interrupt is enabled.

Bit 0 – TRCPT0 Transfer Complete bank 0 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete 1 interrupt.
0.2.4 Device Registers - Endpoint RAM

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled.

13.25.8.4 Device Registers - Endpoint RAM

13.25.8.4.1 Endpoint Descriptor Structure



13.25.8.4.2 Address of Data Buffer

Name: ADDR
Offset: 0x00 & 0x10
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.

13.25.8.4.3 Packet Size

Name: PCKSIZE
Offset: 0x04 & 0x14
Reset: 0xxxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTE_COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	x

Bit 31 – AUTO_ZLP Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the endpoint.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Endpoint size
These bits contains the maximum packet size of the endpoint.

Value	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1023 Byte(1)

(1) for Isochronous endpoints only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multiple Packet Size
These bits define the 14-bit value that is used for multi-packet transfers.
For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.
For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:0 – BYTE_COUNT[13:0] Byte Count
These bits define the 14-bit value that is used for the byte count.
For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.
For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.

13.25.8.4.4 Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xxxxxxxx
Property: NA

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0] VARIABLE
These bits define the VARIABLE field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the VARIABLE field should be read as described below.

VARIABLES	Description
VARIABLE[3:0]	bLinkState (1)
VARIABLE[7:4]	BESL (2)
VARIABLE[8]	bRemoteWake (1)
VARIABLE[10:9]	Reserved

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".
2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0] SUBPID
These bits define the SUBPID field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

13.25.8.4.5 Device Status Bank

Name: STATUS_BK
Offset: 0x0A & 0x1A
Reset: 0xxxxxxxx
Property: NA

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status
This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For OUT transfer, a NAK handshake has been sent.
For Isochronous OUT transfer, an overrun condition has occurred.
For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error
This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.
0.2.5 Host Registers - Common

Value	Description
0	No CRC Error.
1	CRC Error detected.

13.25.8.5 Host Registers - Common

13.25.8.5.1 Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					L1RESUME	VBUSOK	BUSRESET	SOFE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SPDCONF[1:0]		RESUME	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit 11 – L1RESUME Send USB L1 Resume
Writing 0 to this bit has no effect.
1: Generates a USB L1 Resume on the USB bus. This bit should only be set when the Start-of-Frame generation is enabled (SOFE bit set). The duration of the USB L1 Resume is defined by the EXTREG.VARIABLE[7:4] bits field also known as BESL (See LPM ECN).See also 13.25.8.7.4 EXTREG Register.
This bit is cleared when the USB L1 Resume has been sent or when a USB reset is requested.

Bit 10 – VBUSOK VBUS is OK
This notifies the USB HOST that USB operations can be started. When this bit is zero and even if the USB HOST is configured and enabled, HOST operation is halted. Setting this bit will allow HOST operation when the USB is configured and enabled.

Value	Description
0	The USB module is notified that the VBUS on the USB line is not powered.
1	The USB module is notified that the VBUS on the USB line is powered.

Bit 9 – BUSRESET Send USB Reset

Value	Description
0	Reset generation is disabled. It is written to zero when the USB reset is completed or when a device disconnection is detected. Writing zero has no effect.
1	Generates a USB Reset on the USB bus.

Bit 8 – SOFE Start-of-Frame Generation Enable

Value	Description
0	The SOF generation is disabled and the USB bus is in suspend state.
1	Generates SOF on the USB bus in full speed and keep it alive in low speed mode. This bit is automatically set at the end of a USB reset (INTFLAG.RST) or at the end of a downstream resume (INTFLAG.DNRSM) or at the end of L1 resume.

Bits 3:2 – SPDCONF[1:0] Speed Configuration for Host
These bits select the host speed configuration as shown below

Value	Description
0x0	Low and Full Speed capable
0x1	Reserved
0x2	Reserved
0x3	Reserved

Bit 1 – RESUME Send USB Resume
Writing 0 to this bit has no effect.
1: Generates a USB Resume on the USB bus.

This bit is cleared when the USB Resume has been sent or when a USB reset is requested.

13.25.8.5.2 Host Start-of-Frame Control

Name: HSOFC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

During a very short period just before transmitting a Start-of-Frame, this register is locked. Thus, after writing, it is recommended to check the register value, and write this register again if necessary. This register is cleared upon a USB reset.

Bit	7	6	5	4	3	2	1	0
	FLENCE				FLENC[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – FLENCE Frame Length Control Enable
When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].
Note: In Low Speed mode, FLENCE must be '0'.

Value	Description
0	Start-of-Frame is generated every 1ms.
1	Start-of-Frame generation depends on the signed value of FLENC[3:0]. USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms

Bits 3:0 – FLENC[3:0] Frame Length Control
These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.

13.25.8.5.3 Status

Name: STATUS
Offset: 0x0C
Reset: 0x0000
Property: Read only

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	0			0	0		

Bits 7:6 – LINESTATE[1:0] USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0] Speed Status
These bits define the current speed used by the host.

SPEED[1:0]	Speed Status
0x0	Full-speed mode
0x1	Low-speed mode
0x2	Reserved
0x3	Reserved

13.25.8.5.4 Host Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			FNUM[10:5]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bits 13:3 – FNUM[10:0] Frame Number
These bits contains the current SOF number.
These bits can be written by software to initialize a new frame number value. In this case, at the next SOF, the FNUM field takes its new value.
As the FNUM register lies across two consecutive byte addresses, writing byte-wise (8-bits) to the FNUM register may produce incorrect frame number generation. It is recommended to write FNUM register word-wise (32-bits) or half-word-wise (16-bits).

13.25.8.5.5 Host Frame Length

Name: FLENHIGH
Offset: 0x12
Reset: 0x0000
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
	FLENHIGH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FLENHIGH[7:0] Frame Length
These bits contains the 8 high-order bits of the internal frame counter.

Table 13-87. Counter Description vs. Speed

Host Register STATUS.SPEED	Description
Full Speed	With a USB clock running at 12MHz, counter length is 12000 to ensure a SOF generation every 1 ms.

13.25.8.5.6 Host Interrupt Enable Register Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Disconnection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled and an interrupt request will be generated when the Device Disconnection interrupt Flag is set.

Bit 8 – DCONN Device Connection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Connection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled and an interrupt request will be generated when the Device Connection interrupt Flag is set.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM Upstream Resume from Device Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – DNRSM Down Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Down Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Down Resume interrupt is disabled.
1	The Down Resume interrupt is enabled and an interrupt request will be generated when the Down Resume interrupt Flag is set.

Bit 4 – WAKEUP Wake Up Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – RST BUS Reset Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Bus Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled and an interrupt request will be generated when the Bus Reset interrupt Flag is set.

Bit 2 – HSOF Host Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Host Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Host Start-of-Frame interrupt Flag is set.

13.25.8.5.7 Host Interrupt Enable Register Set

Name: INTENSET
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Disconnection interrupt bit and enable the DDSIC interrupt.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled.

Bit 8 – DCONN Device Connection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Connection interrupt bit and enable the DCONN interrupt.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access interrupt bit and enable the RAMACER interrupt.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM Upstream Resume from the device Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume interrupt bit and enable the UPRSM interrupt.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – DNRSM Down Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Down Resume interrupt Enable bit and enable the DNRSM interrupt.

Value	Description
0	The Down Resume interrupt is disabled.
1	The Down Resume interrupt is enabled.

Bit 4 – WAKEUP Wake Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the WAKEUP interrupt request.

Value	Description
0	The WakeUp interrupt is disabled.
1	The WakeUp interrupt is enabled.

Bit 3 – RST Bus Reset Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Bus Reset interrupt Enable bit and enable the Bus RST interrupt.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled.

Bit 2 – HSOF Host Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Host Start-of-Frame interrupt Enable bit and enable the HSOF interrupt.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled.

13.25.8.5.8 Host Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the device has been removed from the USB Bus and will generate an interrupt if INTENCLR/SET.DDISC is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DDISC Interrupt Flag.

Bit 8 – DCONN Device Connection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a new device has been connected to the USB BUS and will generate an interrupt if INTENCLR/SET.DCONN is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DCONN Interrupt Flag.

Bit 7 – RAMACER RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access error occurs during an OUT stage and will generate an interrupt if INTENCLR/SET.RAMACER is one.
Writing a zero to this bit has no effect.

Bit 6 – UPRSM Upstream Resume from the Device Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has received an Upstream Resume signal from the Device and will generate an interrupt if INTENCLR/SET.UPRSM is one.
Writing a zero to this bit has no effect.

Bit 5 – DNRSM Down Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has sent a Down Resume and will generate an interrupt if INTENCLR/SET.DRSM is one.
Writing a zero to this bit has no effect.

Bit 4 – WAKEUP Wake Up Interrupt Flag
This flag is cleared by writing a one.
This flag is set when:
l The host controller is in suspend mode (SOFE is zero) and an upstream resume from the device is detected.
l The host controller is in suspend mode (SOFE is zero) and an device disconnection is detected.
l The host controller is in operational state (VBUSOK is one) and an device connection is detected.
In all cases it will generate an interrupt if INTENCLR/SET.WAKEUP is one.
Writing a zero to this bit has no effect.

Bit 3 – RST Bus Reset Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Bus “Reset” has been sent to the Device and will generate an interrupt if INTENCLR/SET.RST is one.
Writing a zero to this bit has no effect.

Bit 2 – HSOF Host Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB “Host Start-of-Frame” in Full Speed/High Speed or a keep-alive in Low Speed has been sent (every 1 ms) and will generate an interrupt if INTENCLR/SET.HSOF is one.
The value of the FNUM register is updated.
Writing a zero to this bit has no effect.

13.25.8.5.9 Pipe Interrupt Summary

Name: PINTSMRY
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	PINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PINT[15:0]
The flag PINT[n] is set when an interrupt is triggered by the pipe n. See 13.25.8.6.6 PINTFLAG register in the Host Pipe Register section.
This bit will be cleared when there are no interrupts pending for Pipe n.
Writing to this bit has no effect.

13.25.8.6 Host Registers - Pipe

13.25.8.6.1 Host Pipe n Configuration

Name: PCFGn
Offset: 0x100 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			PTYPE[2:0]			BK	PTOKEN[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:3 – PTYPE[2:0] Type of the Pipe
These bits contains the pipe type.

PTYPE[2:0]	Description
0x0	Pipe is disabled
0x1	Pipe is enabled and configured as CONTROL
0x2	Pipe is enabled and configured as ISO
0x3	Pipe is enabled and configured as BULK
0x4	Pipe is enabled and configured as INTERRUPT
0x5	Pipe is enabled and configured as EXTENDED
0x06-0x7	Reserved

These bits are cleared upon sending a USB reset.

Bit 2 – BK Pipe Bank
This bit selects the number of banks for the pipe.
For control endpoints writing a zero to this bit is required as only Bank0 is used for Setup/In/Out transactions.
This bit is cleared when a USB reset is sent.

BK(1)	Description
0x0	Single-bank endpoint
0x1	Dual-bank endpoint

1. Bank field is ignored when PTYPE is configured as EXTENDED.

Value	Description
0	A single bank is used for the pipe.
1	A dual bank is used for the pipe.

Bits 1:0 – PTOKEN[1:0] Pipe Token
These bits contains the pipe token.

PTOKEN[1:0](1)	Description
0x0	SETUP(2)
0x1	IN
0x2	OUT
0x3	Reserved

1. PTOKEN field is ignored when PTYPE is configured as EXTENDED.
2. Available only when PTYPE is configured as CONTROL

Theses bits are cleared upon sending a USB reset.

13.25.8.6.2 Interval for the Bulk-Out/Ping Transaction

Name: BINTERVAL
Offset: 0x103 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BINTERVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BINTERVAL[7:0] BINTERVAL
These bits contains the Ping/Bulk-out period.
These bits are cleared when a USB reset is sent or when PEN[n] is zero.

BINTERVAL	Description
=0	Multiple consecutive OUT token is sent in the same frame until it is acked by the peripheral
>0	One OUT token is sent every BINTERVAL frame until it is acked by the peripheral

Depending from the type of pipe the desired period is defined as:

PTYPE	Description
Interrupt	1 ms to 255 ms
Isochronous	2^(Binterval) * 1 ms
Bulk or control	1 ms to 255 ms
EXT LPM	bInterval ignored. Always 1 ms when a NYET is received.

13.25.8.6.3 Pipe Status Clear n

Name: PSTATUSCLR
Offset: 0x104 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R/W	R/W		R/W		R/W		R/W
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK0RDY bit.

Bit 4 – PFREEZE Pipe Freeze Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.PFREEZE bit.

Bit 2 – CURBK Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.CURBK bit.

Bit 0 – DTGL Data Toggle Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.DTGL bit.

13.25.8.6.4 Pipe Status Set Register n

Name: PSTATUSSET
Offset: 0x105 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R/W	R/W		R/W		R/W		R/W
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK1RDY.

Bit 6 – BK0RDY Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK0RDY.

Bit 4 – PFREEZE Pipe Freeze Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.PFREEZE bit.

Bit 2 – CURBK Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.CURBK bit.

Bit 0 – DTGL Data Toggle Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.DTGL bit.

13.25.8.6.5 Pipe Status Register n

Name: PSTATUS
Offset: 0x106 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R	R		R		R		R
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 is ready
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.
This bank is not used for Control pipe.

Value	Description
0	The bank number 1 is not ready: For IN the bank is empty. For Control/OUT the bank is not yet fill in.
1	The bank number 1 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 6 – BK0RDY Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.
This bank is the only one used for Control pipe.

Value	Description
0	The bank number 0 is not ready: For IN the bank is not empty. For Control/OUT the bank is not yet fill in.
1	The bank number 0 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 4 – PFREEZE Pipe Freeze
Writing a one to the bit EPSTATUSCLR.PFREEZE will clear this bit.
Writing a one to the bit EPSTATUSSET.PFREEZE will set this bit.
This bit is also set by the hardware:

- When a STALL handshake has been received.
- After a PIPE has been enabled (rising of bit PEN.N).
- When an LPM transaction has completed whatever handshake is returned or the transaction was timed-out.
- When a pipe transfer was completed with a pipe error. See 13.25.8.6.6 PINTFLAG register.

When PFREEZE bit is set while a transaction is in progress on the USB bus, this transaction will be properly completed. PFREEZE bit will be read as “1” only when the ongoing transaction will have been completed.

Value	Description
0	The Pipe operates in normal operation.
1	The Pipe is frozen and no additional requests will be sent to the device on this pipe address.

Bit 2 – CURBK Current Bank

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 0 – DTGL Data Toggle Sequence
Writing a one to the bit EPSTATUSCLR.DTGL will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGL will set this bit.
This bit is toggled automatically by hardware after a data transaction.
This bit will reflect the data toggle in regards of the token type (IN/OUT/SETUP).

Value	Description
0	The PID of the next expected transaction will be zero: data 0.

Value	Description
1	The PID of the next expected transaction will be one: data 1.

13.25.8.6.6 Host Pipe Interrupt Flag Register

Name: PINTFLAG
Offset: 0x107 + (n x 0x20)
Reset: 0x0000
Property: -

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL STALL Received Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a stall occurs and will generate an interrupt if PINTENCLR/SET.STALL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the STALL Interrupt Flag.

Bit 4 – TXSTP Transmitted Setup Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Complete occurs and will generate an interrupt if PINTENCLR/SET.TXSTP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TXSTP Interrupt Flag.

Bit 3 – PERR Pipe Error Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a pipe error occurs and will generate an interrupt if PINTENCLR/SET.PERR is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the PERR Interrupt Flag.

Bit 2 – TRFAIL Transfer Fail Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Fail occurs and will generate an interrupt if PINTENCLR/SET.TRFAIL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRFAIL Interrupt Flag.

Bit 1 – TRCPT1 Transfer Complete 1 interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT1 is one.
PINTFLAG.TRCPT1 is set for a double bank IN/OUT pipe when current bank is 1.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT1 Interrupt Flag.

Bit 0 – TRCPT0 Transfer Complete 0 interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT0 is one.
PINTFLAG.TRCPT0 is set for a single bank IN/OUT pipe or a double bank IN/OUT pipe when current bank is 0.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT0 Interrupt Flag.

13.25.8.6.7 Host Pipe Interrupt Clear Register

Name: PINTENCLR
Offset: 0x108 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENSET) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL Received Stall Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Stall interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The received Stall interrupt is disabled.
1	The received Stall interrupt is enabled and an interrupt request will be generated when the received Stall interrupt Flag is set.

Bit 4 – TXSTP Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmitted Setup interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled and an interrupt request will be generated when the Transmitted Setup interrupt Flag is set.

Bit 3 – PERR Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Pipe Error interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled and an interrupt request will be generated when the Pipe Error interrupt Flag is set.

Bit 2 – TRFAIL Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled and an interrupt request will be generated when the Transfer Fail interrupt Flag is set.

Bit 1 – TRCPT1 Transfer Complete Bank 1 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete interrupt Enable bit 1 and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete Bank 1 interrupt is disabled.

Value	Description
1	The Transfer Complete Bank 1 interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt Flag 1 is set.

Bit 0 – TRCPT0 Transfer Complete Bank 0 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete interrupt Enable bit 0 and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete Bank 0 interrupt is disabled.
1	The Transfer Complete Bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt 0 Flag is set.

13.25.8.6.8 Host Interrupt Pipe Set Register

Name: PINTENSET
Offset: 0x109 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENCLR) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL Stall Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Stall interrupt.

Value	Description
0	The Stall interrupt is disabled.
1	The Stall interrupt is enabled.

Bit 4 – TXSTP Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmitted Setup interrupt.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled.

Bit 3 – PERR Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Pipe Error interrupt.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled.

Bit 2 – TRFAIL Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 1 – TRCPT1 Transfer Complete 1 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete interrupt Enable bit 1.

Value	Description
0	The Transfer Complete 1 interrupt is disabled.
1	The Transfer Complete 1 interrupt is enabled.

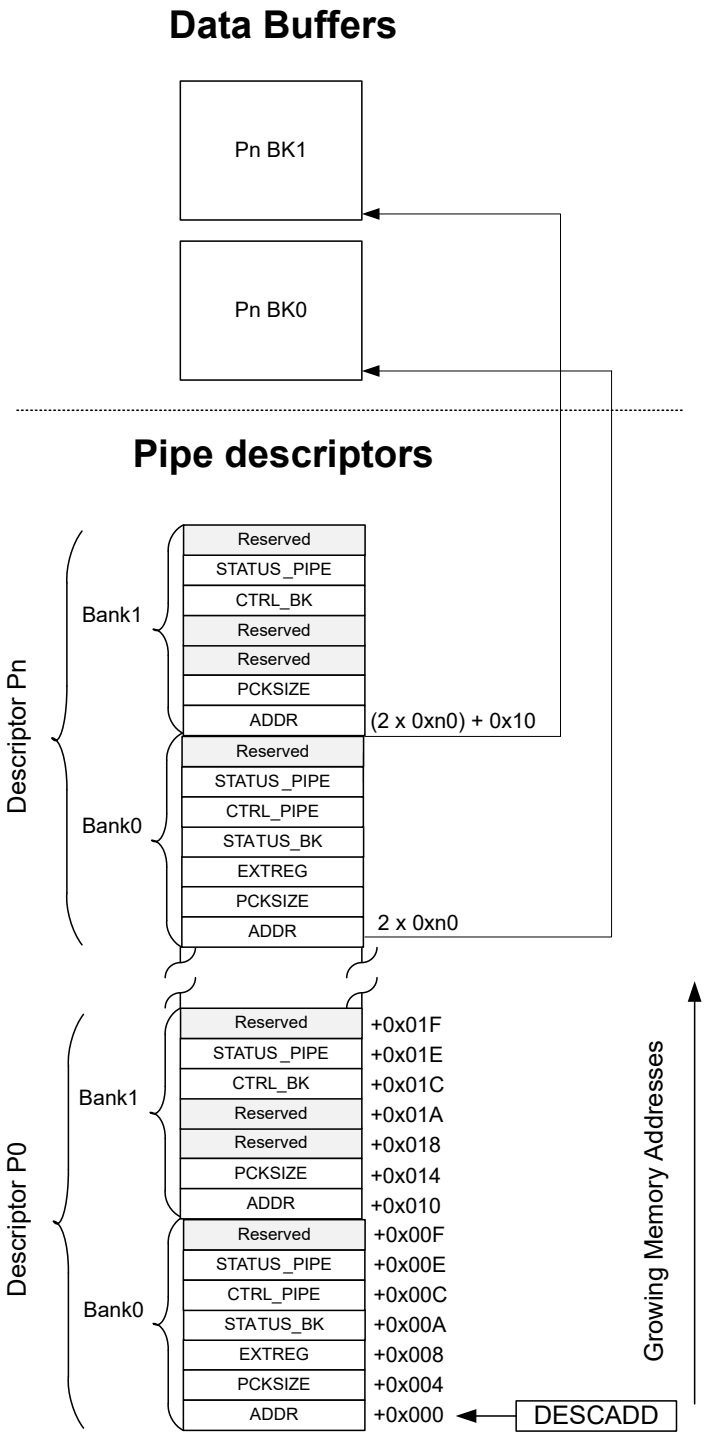
Bit 0 – TRCPT0 Transfer Complete 0 interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete interrupt Enable bit 0.
0.2.7 Host Registers - Pipe RAM

Value	Description
0	The Transfer Complete 0 interrupt is disabled.

Value	Description
1	The Transfer Complete 0 interrupt is enabled.

13.25.8.7 Host Registers - Pipe RAM

13.25.8.7.1 Pipe Descriptor Structure



13.25.8.7.2 Address of the Data Buffer

Name: ADDR
Offset: 0x00 & 0x10
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute double word address in RAM. The two least significant bits must be zero to ensure the descriptor is 32-bit aligned.

13.25.8.7.3 Packet Size

Name: PCKSIZE
Offset: 0x04 & 0x14
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – AUTO_ZLP Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the pipe.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for OUT pipes only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Pipe size
These bits contains the size of the pipe.
Theses bits are cleared upon sending a USB reset.

SIZE[2:0]	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1024 Byte in HS mode(1) 1023 Byte in FS mode(1)

1. For Isochronous pipe only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multi Packet IN or OUT size
These bits define the 14-bit value that is used for multi-packet transfers.
For IN pipes, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT pipes, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:8 – BYTE_COUNT[5:0] Byte Count
These bits define the 14-bit value that contains number of bytes sent in the last OUT or SETUP transaction for an OUT pipe, or of the number of bytes to be received in the next IN transaction for an input pipe.

13.25.8.7.4 Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xxxxxxxx
Property: NA

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0] Extended variable
These bits define the VARIABLE field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum.”
To support the USB2.0 Link Power Management addition the VARIABLE field should be set as described below.

VARIABLE	Description
VARIABLE[3:0]	bLinkState(1)
VARIABLE[7:4]	BESL (See LPM ECN)(2)
VARIABLE[8]	bRemoteWake(1)
VARIABLE[10:9]	Reserved

(1) for a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum"
(2) for a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0] SUBPID
These bits define the SUBPID field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the SUBPID field should be set as described in “Table 2.2 SubPID Types in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

13.25.8.7.5 Host Status Bank

Name: STATUS_BK
Offset: 0x0A
Reset: 0xXXXXXXX
Property: NA

Original offset 0x0A & 0x1A

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status
This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error
This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

13.25.8.7.6 Host Control Pipe

Name: CTRL_PIPE
Offset: 0x0C
Reset: 0xxxxxxxx
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERMAX[3:0]				PEPNUM[3:0]			
Access	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bit	7	6	5	4	3	2	1	0
		PDADDR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	x

Bits 15:12 – PERMAX[3:0] Pipe Error Max Number
These bits define the maximum number of error for this Pipe before freezing the pipe automatically.

Bits 11:8 – PEPNUM[3:0] Pipe EndPoint Number
These bits define the number of endpoint for this Pipe.

Bits 6:0 – PDADDR[6:0] Pipe Device Address
These bits define the Device Address for this pipe.

13.25.8.7.7 Host Status Pipe

Name: STATUS_PIPE
Offset: 0x0E
Reset: 0xXXXXXXX
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Original offset 0x0E & 0x1E

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	x	x	x	x	x	x

Bits 7:5 – ERCNT[2:0] Pipe Error Counter
These bits define the number of errors detected on the pipe.

Bit 4 – CRC16ER CRC16 ERROR
This bit defines the CRC16 Error Status.
This bit is set when a CRC 16 error has been detected during a IN transactions.

Value	Description
0	No CRC 16 Error detected.
1	A CRC 16 error has been detected.

Bit 3 – TOUTER TIME OUT ERROR
This bit defines the Time Out Error Status.
This bit is set when a Time Out error has been detected during a USB transaction.

Value	Description
0	No Time Out Error detected.
1	A Time Out error has been detected.

Bit 2 – PIDER PID ERROR
This bit defines the PID Error Status.
This bit is set when a PID error has been detected during a USB transaction.

Value	Description
0	No PID Error detected.
1	A PID error has been detected.

Bit 1 – DAPIDER Data PID ERROR
This bit defines the PID Error Status.
This bit is set when a Data PID error has been detected during a USB transaction.

Value	Description
0	No Data PID Error detected.
1	A Data PID error has been detected.

Bit 0 – DTGLER Data Toggle Error
This bit defines the Data Toggle Error Status.
This bit is set when a Data Toggle Error has been detected.

Value	Description
0	No Data Toggle Error.
1	Data Toggle Error detected.

13.26 CCL – Configurable Custom Logic

13.26.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each LookUp Table (LUT) consists of three inputs, a truth table, and as options synchronizer, filter and edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

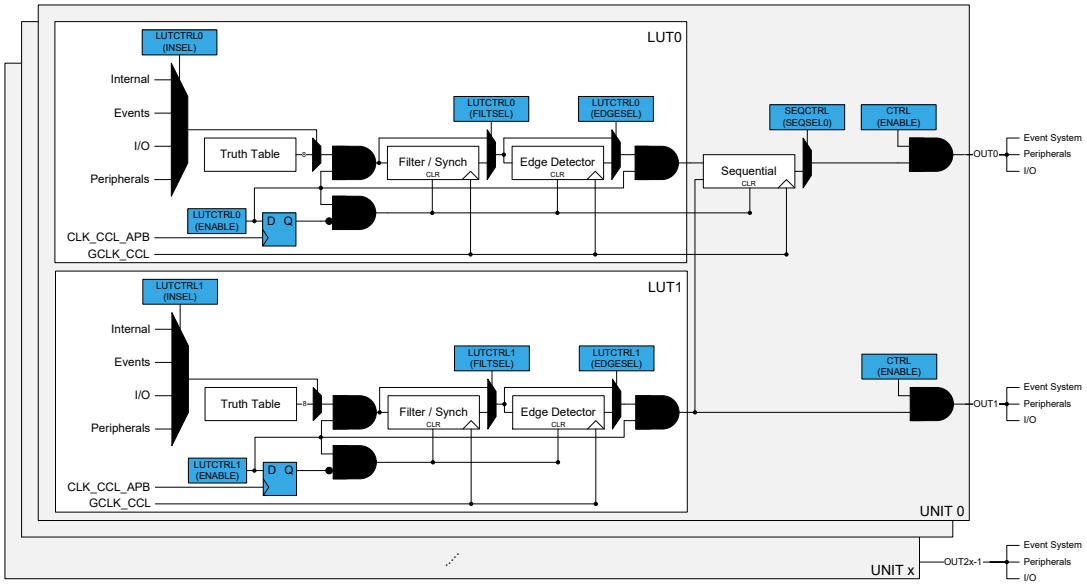
The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

13.26.2 Features

- Glue logic for general purpose PCB design
- Up to four Programmable LookUp Table (LUT)
- Combinatorial Logic Functions:
  AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential Logic Functions:
  Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LookUp Table Inputs Selection:
  - I/Os
  - Events
  - Internal Peripherals
  - Subsequent LUT Output
- Output can be connected to IO pins or Event System
- Optional synchronizer, filter, or edge detector available on each LUT output

13.26.3 Block Diagram

Figure 13-186. Configurable Custom Logic



13.26.4 Signal Description

Pin Name	Type	Description
OUT[n:0]	Digital output	Output from lookup table
IN[3n+2:0]	Digital input	Input to lookup table

1. n is the number of CCL groups.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.26.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.26.5.1 I/O Lines

Using the CCL I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

13.17 PORT - I/O Pin Controller

13.26.5.2 Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

13.8 PM – Power Manager

13.26.5.3 Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the power manager, and the default state of CLK_CCL_APB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the sequential sub-module of CCL. GCLK_CCL is required when input events, a filter, an edge detector, or asequential sub-module is enabled. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

Related Links

13.6.6.2.6 Peripheral Clock Masking

13.5 GCLK - Generic Clock Controller

13.26.5.4 DMA

Not applicable.

13.26.5.5 Interrupts

Not applicable.

13.26.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

13.18 EVSYS – Event System

13.26.5.7 Debug Operation

When the CPU is halted in debug mode the CCL continues normal operation. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

13.26.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.26.5.9 Analog Connections

Not applicable.

13.26.6 Functional Description

13.26.6.1 Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

13.26.6.2 Basic Operation

13.26.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

13.26.6.2.2 Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to 13.26.8.1 CTRL for details.

13.26.6.2.3 Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 13-187. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 13-187. Truth Table Output Value Selection

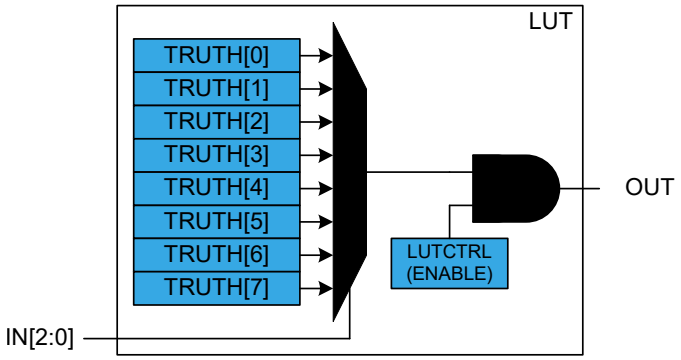


Table 13-88. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

13.26.6.2.4 Truth Table Inputs Selection

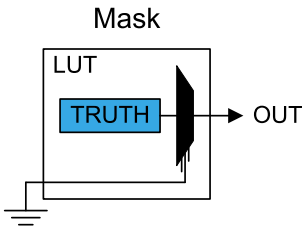
Input Overview
The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input y of LUT x is configured by writing the Input y Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)
When a LUT input is masked (LUTCTRLx.INSELy=MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 13-188. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)
When selected (LUTCTRLx.INSELy=FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

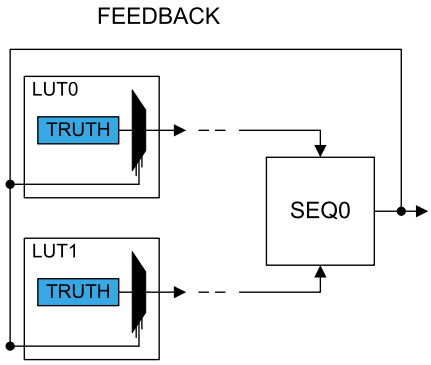
$\mathrm{IN}[2\mathrm{N}][i] = \mathrm{SEQ}[N]$

$\mathrm{IN}[2\mathrm{N}+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

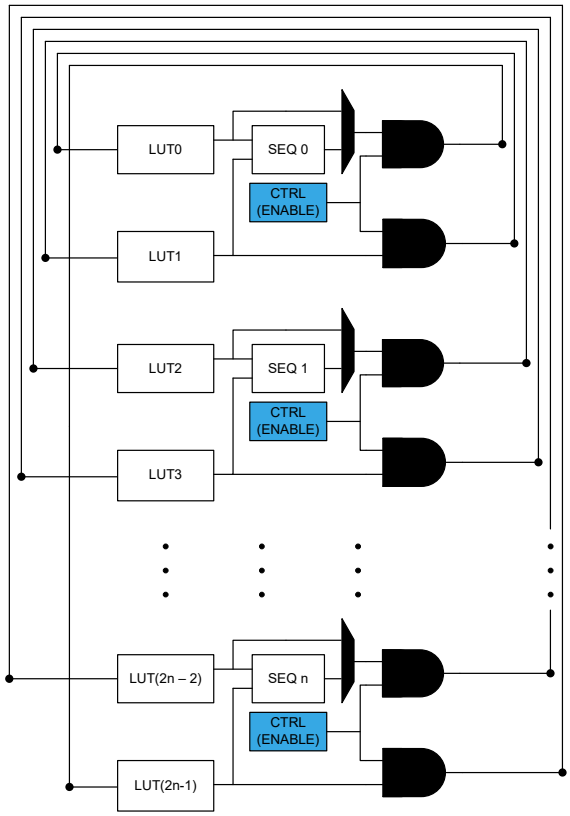
For details, refer to 13.26.6.2.7 Sequential Logic.

Figure 13-189. Feedback Input Selection



Linked LUT (LINK)
When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (e.g., LUT2 is the input for LUT1), as shown in this figure:

Figure 13-190. Linked LUT Input Selection



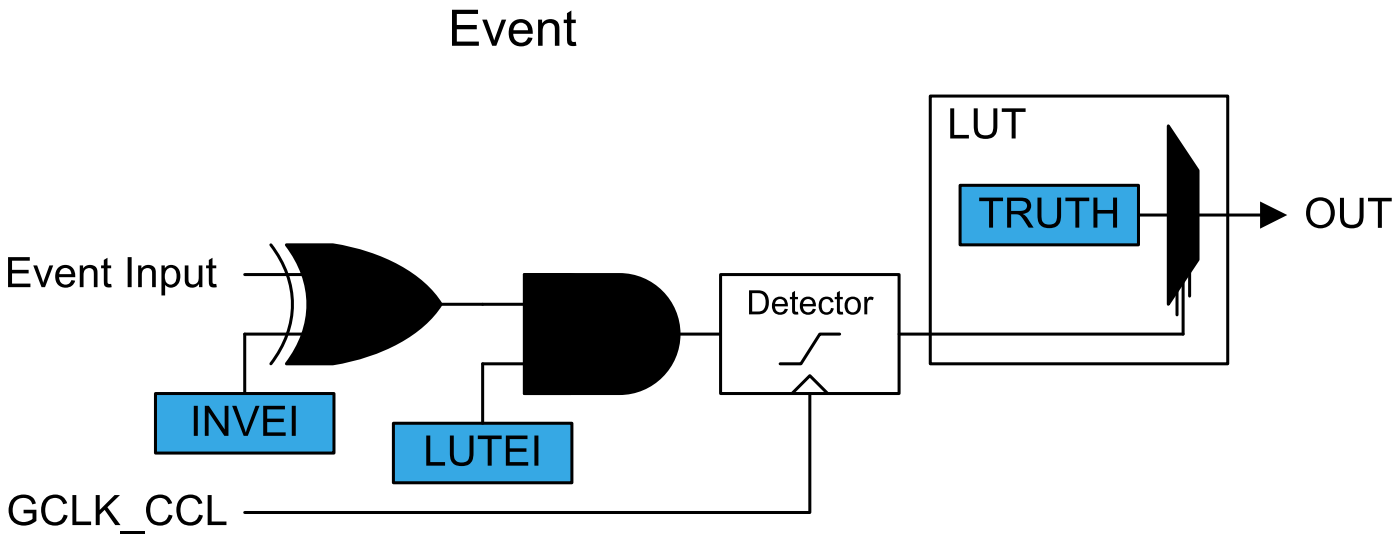
Internal Events Inputs Selection (EVENT)

Asynchronous events from the Event System can be used as input selection, as shown in Figure 13-191. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle. Writing the LUTCTRLx.INSELy=ASYNCEVENT will disable the edge detector. In this case, it is possible to combine an asynchronous event input with any other input source. This is typically useful with event levels inputs (external IO pin events, as example). The following steps ensure proper operation:

1. Enable the GCLK_CCL clock.
2. Configure the Event System to route the event asynchronously.
3. Select the event input type (LUTCTRLx.INSEL).
4. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLx.INVEI) .
5. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLx.LUTEI) to '1'.

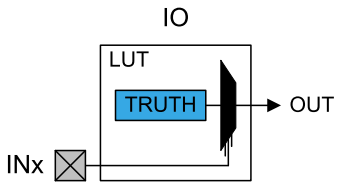
Figure 13-191. Event Input Selection



I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLx.INSELy=IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 13-192. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

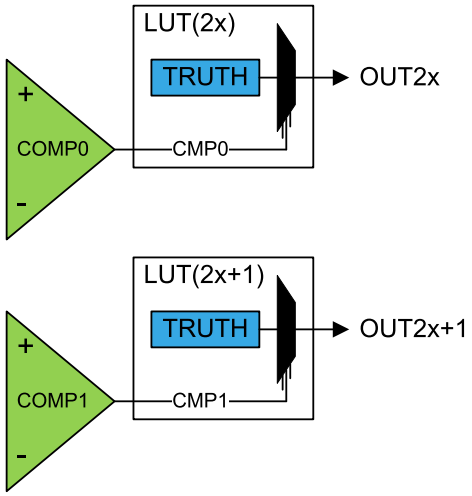
```
IN[N][i]=AC[N % ComparatorOutput_Number]
```

With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.

Figure 13-193. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLx.INSELy=TC). Only consecutive instances of the TC, i.e. TCx and the subsequent TC(x+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC\_Instance\_Number}]$

$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC\_Instance\_Number}]$

Where N represents the LUT number and i represents the LUT input index (i=0,1,2).

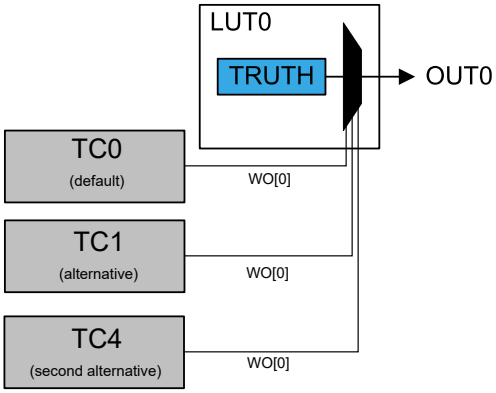
For devices with more than four TC instances, it is also possible to enable a second alternative option (LUTCTRLx.INSEL=ALT2TC). This option is intended to relax the alternative pin function or PCB design constraints when the default or the alternative TC instances are used for other purposes. When enabled, the Timer/Counter selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = SecondAlternativeTC[(N+4) \,\%\, \mathrm{TC\_Instance\_Number}]$

Note that for not implemented TC_Instance_Number, the corresponding input is tied to ground.

Before selecting the waveform outputs, the TC must be configured first.

Figure 13-194. TC Input Selection



Timer/Counter for Control Application Inputs (TCC)
The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (i.e., IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

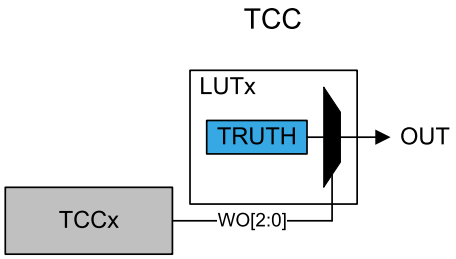
Note:
The TCC selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = TCC[N \,\%\, TCC\_\mathrm{Instance\_Number}]$

Where *N* represents the LUT number.

Before selecting the waveform outputs, the TCC must be configured first.

Figure 13-195. TCC Input Selection



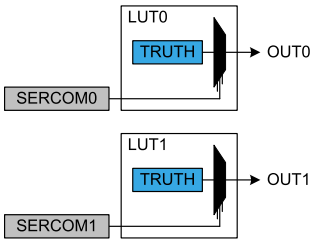
Serial Communication Output Transmit Inputs (SERCOM)
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM\_Instance\_Number}]$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

Figure 13-196. SERCOM Input Selection

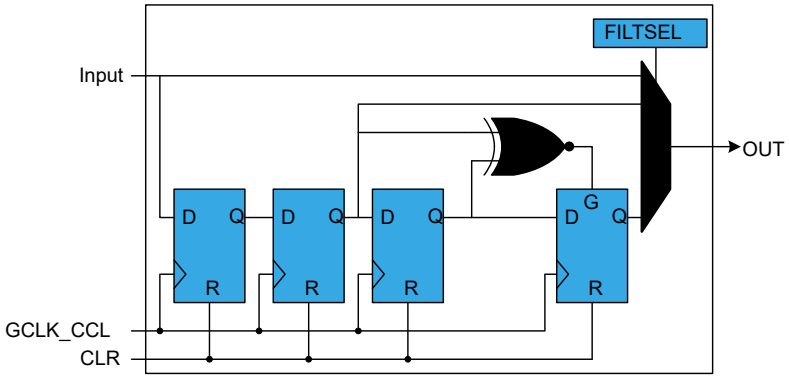


Related Links
7. I/O Multiplexing and Considerations
13.17 PORT - I/O Pin Controller
13.5 GCLK - Generic Clock Controller
13.28 AC – Analog Comparators
13.23 TC – Timer/Counter
13.24 TCC – Timer/Counter for Control Applications
13.19 SERCOM – Serial Communication Interface

13.26.6.2.5 Filter
By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.
Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 13-197. Filter



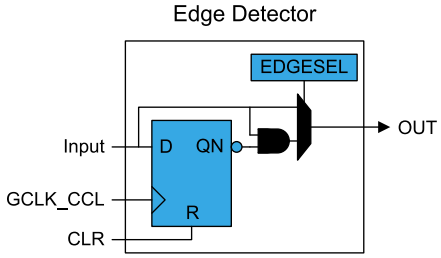
13.26.6.2.6 Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be programmed to provide the opposite levels.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, a valid filter option must be enabled as well.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 13-198. Edge Detector



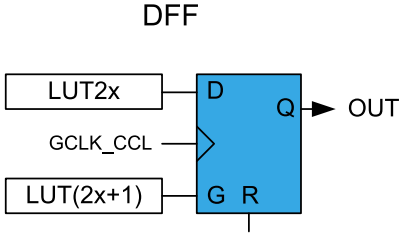
13.26.6.2.7 Sequential Logic

Each LUT pair can be connected to internal sequential logic: D flip flop, JK flip flop, gated D-latch or RS-latch can be selected by writing the corresponding Sequential Selection bits in Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK clock and optionally each LUT filter or edge detector, must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT2x), and the G-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 13-199.

Figure 13-199. D Flip Flop



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 13-89.

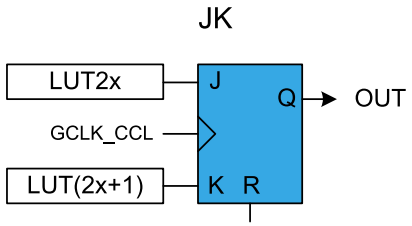
Table 13-89. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT2x), and the K-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 13-200.

Figure 13-200. JK Flip Flop



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 13-90.

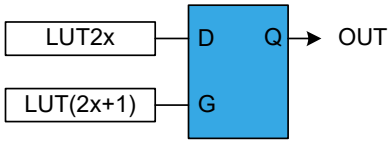
Table 13-90. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT2x), and the G-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 13-199.

Figure 13-201. D-Latch



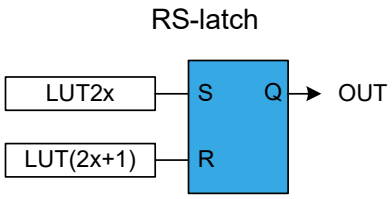
When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 13-91.

Table 13-91. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)
When this configuration is selected, the S-input is driven by the even LUT output (LUT2x), and the R-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 13-202.

Figure 13-202. RS-Latch



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 13-92.

Table 13-92. RS-latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

13.26.6.3 Events

The CCL can generate the following output events:

- LUTOUTx: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuration.

The CCL can take the following actions on an input event:

- INx: The event is used as input for the TRUTH table. For further details refer to 13.26.5.6 Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event. Refer to *EVSYS – Event System* for details on configuration.

Related Links
13.18 EVSYS – Event System

13.26.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in all sleep modes.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

13.26.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL0	7:0					SEQSEL[3:0]			
0x05	SEQCTRL1	7:0					SEQSEL[3:0]			
0x06 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							
0x10	LUTCTRL2	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							
0x14	LUTCTRL3	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							

13.26.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.26.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.26.8.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in standby mode. The setting is ignored for configurations where the generic clock is not required. For details refer to 13.26.6.4 Sleep Mode Operation.

Value	Description
0	Generic clock is not required in standby sleep mode.
1	Generic clock is required in standby sleep mode.

Bit 1 – ENABLE Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.26.8.2 Sequential Control x

Name: SEQCTRL
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Note: SEQCTRLx register is Enable-protected when LUTCTRLx.ENABLE = 1.

Bit	7	6	5	4	3	2	1	0
					SEQSEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – SEQSEL[3:0] Sequential Selection
These bits select the sequential configuration:
Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 – 0xF		Reserved

13.26.8.3 LUT Control x

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Note: The LUTCTRLx register is enable-protected when CCL.LUTCTRLx.ENABLE = 1.

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSEL1[3:0]				INSEL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	R/W		R/W	R/W			R/W	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0] Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bits 8:11, 12:15, 16:19 – INSELx LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	AC	AC input source: CMP[0] (LUT0) / CMP[1] (LUT1)
0x6	TC	TC input source: TC0 (LUT0) / TC1 (LUT1)

Value	Name	Description
0x7	ALTTC	Alternative TC input source: TC1 (LUT0) / TC2 (LUT1)
0x8	Reserved	Reserved
0x9	SERCOM	SERCOM input source: SERCOM0 (LUT0) / SERCOM1 (LUT1)
0xA – 0xF	-	Reserved

Bit 7 – EDGESEL Edge Selection

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0] Filter Selection
These bits select the LUT output filter options:
Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	-	Reserved

Bit 1 – ENABLE LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

13.27 ADC – Analog-to-Digital Converter

13.27.1 Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

An integrated temperature sensor is available for use with the ADC. The bandgap voltage as well as the scaled I/O and core voltages can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

The ADC can be configured for 8-, 10- or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

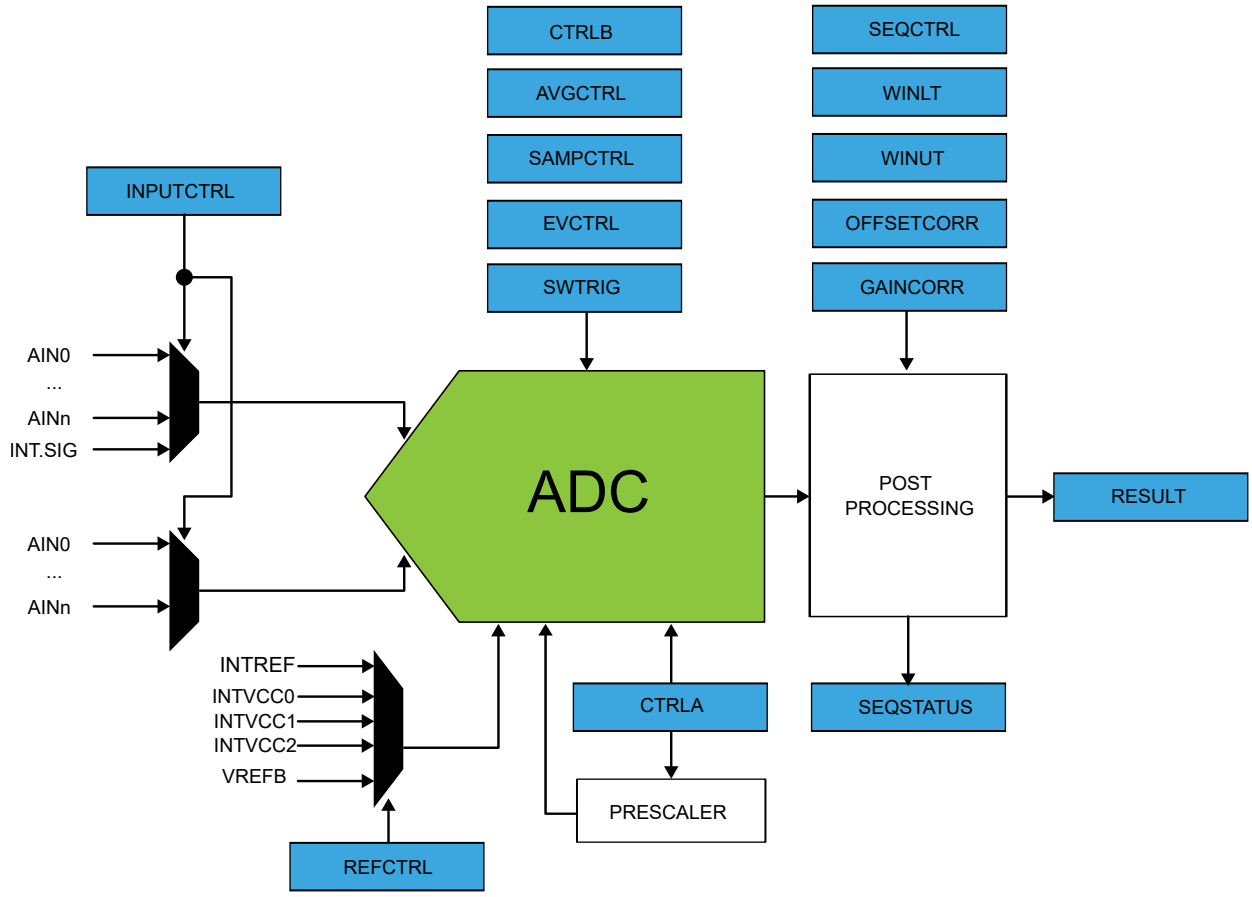
13.27.2 Features

- 8-, 10- or 12-bit resolution
- Up to 1,000,000 samples per second (1 MSPS)
- Differential and single-ended inputs
  - Up to 20 analog inputs
    28 positive and 10 negative, including internal and external
- Internal inputs:
  - Internal temperature sensor

  - Bandgap voltage
  - Scaled core supply
  - Scaled I/O supply
  - Scaled VBAT supply
- Single, continuous and sequencing options
- Windowing monitor with selectable channel
- Conversion range: V_{ref} = [1.0V to VDD_{ANA}]
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)
- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time
- Flexible Power / Throughput rate management

13.27.3 Block Diagram

Figure 13-203. ADC Block Diagram



13.27.4 Signal Description

Signal	Description	Type
VREFB	Analog input	External reference voltage B
AIN[19..0]	Analog input	Analog input channels

Note: One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.27.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.27.5.1 I/O Lines

Using the ADC's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

Related Links

13.17 PORT - I/O Pin Controller

13.27.5.2 Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.27.5.3 Clocks

The ADC bus clock (CLK_APB_ADCx) can be enabled in the Main Clock, which also defines the default state.

The ADC requires a generic clock (GCLK_ADC). This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the ADC.

A generic clock is asynchronous to the bus clock. Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *Synchronization* for further details.

Related Links

13.27.6.8 Synchronization
13.6.6.2.6 Peripheral Clock Masking
13.5 GCLK - Generic Clock Controller

13.27.5.4 DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the ADC DMA requests requires the DMA Controller to be configured first.

Related Links

13.14 DMAC – Direct Memory Access Controller

13.27.5.5 Interrupts

The interrupt request line is connected to the interrupt controller. Using the ADC interrupt requires the interrupt controller to be configured first.

Related Links

11.2 Nested Vector Interrupt Controller

13.27.5.6 Events

The events are connected to the Event System.

Related Links

13.18 EVSYS – Event System

13.27.5.7 Debug Operation

When the CPU is halted in debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to 13.27.8.18 DBGCTRL for details.

13.27.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following register:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.27.5.9 Analog Connections

I/O-pins (AINx) and the VREFB reference voltage pin are analog inputs to the ADC.

13.27.5.10 Calibration

The BIAS and LINEARITY calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to achieve specified accuracy.

Related Links

10.4 NVM Software Calibration Area Mapping

13.27.6 Functional Description

13.27.6.1 Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see 13.27.6.2.8 Conversion Timing and Sampling Rate.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal (e.g., an internal temperature sensor) or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

13.27.6.2 Basic Operation

13.27.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Reference Control register (REFCTRL)
- Event Control register (EVCTRL)
- Calibration register (CALIB)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

13.27.6.2.2 Enabling, Disabling and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to 13.27.8.1 CTRLA for details.

13.27.6.2.3 Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADC frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in Initialization. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

To avoid data loss if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

13.27.6.2.4 Prescaler Selection

The ADC is clocked by GCLK_ADC. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLB for details on prescaler settings. Refer to 13.27.6.2.8 Conversion Timing and Sampling Rate for details on timing and sampling rate.

Figure 13-204. ADC Prescaler

Note: The minimum prescaling factor is DIV2.

13.27.6.2.5 Reference Configuration

The ADC has various sources for its reference voltage V_{REF}. The Reference Voltage Selection bit field in the Reference Control register (REFCTRL.REFSEL) determines which reference is selected. By default, the internal voltage reference INTREF is selected. Based on the customer application requirements, the external or internal reference can be selected. Refer to REFCTRL for further details on available selections.

13.27.6.2.6 ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control C register (CTRLC.RESSEL). By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also 13.27.6.2.8 Conversion Timing and Sampling Rate.

13.27.6.2.7 Differential and Single-Ended Conversions

The ADC has two conversion options: differential and single-ended:

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

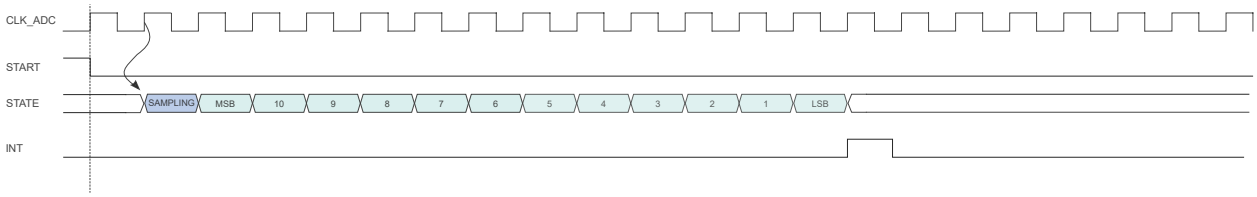
If the positive input may go below the negative input, the differential mode should be used in order to get correct results.

The differential mode is enabled by setting DIFFMODE bit in the Control C register (CTRLC.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

13.27.6.2.8 Conversion Timing and Sampling Rate

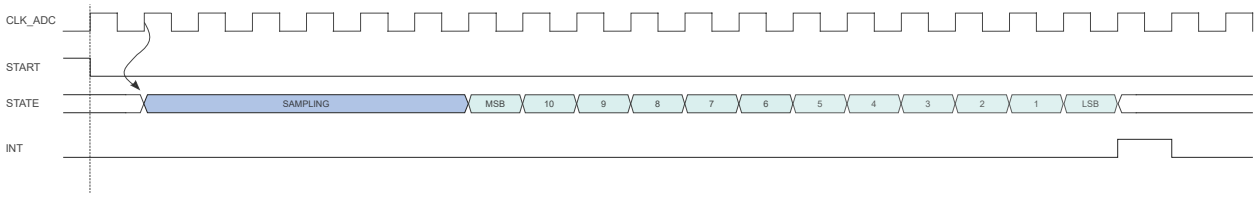
The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADC clock. The input channel is sampled in the first half CLK_ADC period.

Figure 13-205. ADC Timing for One Conversion in 12-bit Resolution



The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). As example, the next figure is showing the timing conversion with sampling time increased to six CLK_ADC cycles.

Figure 13-206. ADC Timing for One Conversion with Increased Sampling Time, 12-bit



The ADC provides also offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

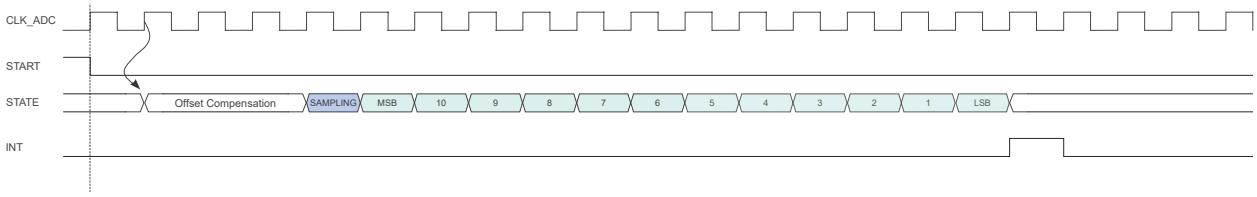
Note: If offset compensation is used, the sampling time must be set to one cycle of CLK_ADC.

In free running mode, the sampling rate R_S is calculated by

$R_S = f_{CLK\_ADC} / (n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

Here, $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles, $n_{OFFCOMP}$ is the offset compensation duration in clock cycles, and n_{DATA} is the bit resolution. f_{CLK_ADC} is the ADC clock frequency from the internal prescaler: $f_{CLK\_ADC} = f_{GCLK\_ADC} / 2^{(1 + CTRLA.PRESCALER)}$

Figure 13-207. ADC Timing for One Conversion with Offset Compensation, 12-bit



The impact of resolution on the sampling rate is seen in the next two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

Figure 13-208. ADC Timing for Free Running in 12-bit Resolution

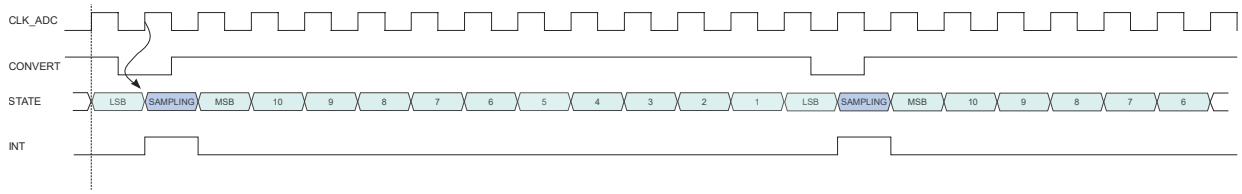
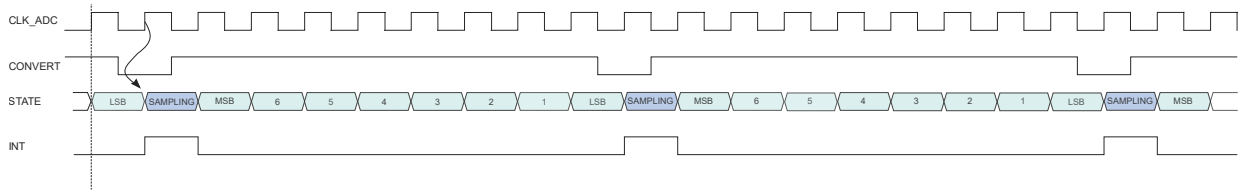


Figure 13-209. ADC Timing for Free Running in 8-bit Resolution



The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

> **Example.** In order to obtain 1MSPS in 12-bit resolution with a sampling time length of four CLK_ADC cycles, f_{CLK_ADC} must be 1MSPS * (4 + 12) = 16MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32MHz.

13.27.6.2.9 Accumulation

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to match the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

Note: To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Table 13-93. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	0	12 bits	0
2	0x1	0	13 bits	0
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

13.27.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in 13.27.6.2.9 Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 13-94.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 13-94.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 13-94. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2
64	0x6	18	2	16	0x4	6	12 bits	4

.....continued

Number of Accumulated Samples	AVGCTRL.SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
128	0x7	19	3	16	0x4	7	12 bits	8
256	0x8	20	4	16	0x4	8	12 bits	16
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

13.27.6.2.11 Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

To increase the resolution by n bits, 4n samples must be accumulated. The result must then be right-shifted by n bits. This right-shift is a combination of the automatic right-shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in n bit extra LSB resolution.

Table 13-95. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

13.27.6.2.12 Automatic Sequences

The ADC has the ability to automatically sequence a series of conversions. This means that each time the ADC receives a start-of-conversion request, it can perform multiple conversions automatically. All of the 32 positive inputs can be included in a sequence by writing to corresponding bits in the Sequence Control register (SEQCTRL). The order of the conversion in a sequence is the lower positive MUX selection to upper positive MUX (AIN0, AIN1, AIN2 ...). In differential mode, the negative inputs selected by MUXNEG field, will be used for the entire sequence.

When a sequence starts, the Sequence Busy status bit in Sequence Status register (SEQSTATUS.SEQBUSY) will be set. When the sequence is complete, the Sequence Busy status bit will be cleared.

Each time a conversion is completed, the Sequence State bit in Sequence Status register (SEQSTATUS.SEQSTATE) will store the input number from which the conversion is done. The result will be stored in the RESULT register, and the Result Ready Interrupt Flag (INTFLAG.RESRDY) is set.

If additional inputs must be scanned, the ADC will automatically start a new conversion on the next input present in the sequence list.

Note that if SEQCTRL register has no bits set to '1', the conversion is done with the selected MUXPOS input.

13.27.6.2.13 Window Monitor

The window monitor feature allows the conversion result in the RESULT register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control C register (CTRLC.RESSEL). This means that for example in 8-bit mode,

only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

13.27.6.2.14 Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

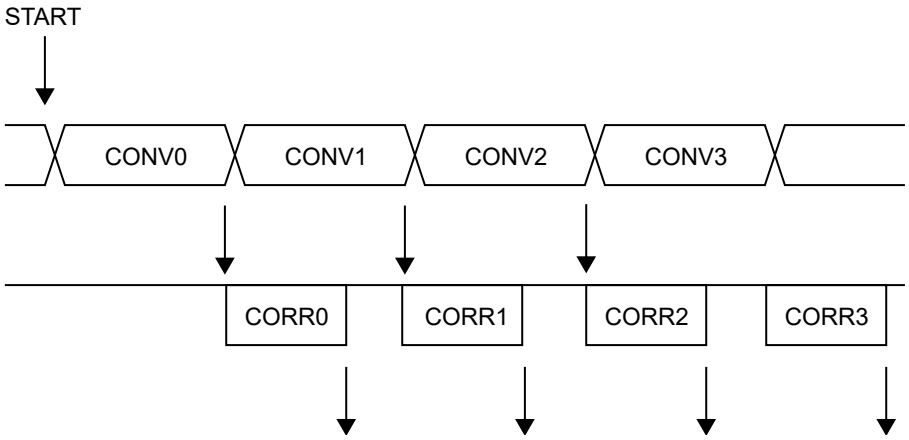
To correct these two errors, the Digital Correction Logic Enabled bit in the Control C register (CTRLC.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + - \text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

Figure 13-210. ADC Timing Correction Enabled



13.27.6.3 Additional Features

13.27.6.3.1 Double Buffering

The following registers are double buffered:

- Input Control (INPUTCTRL)
- Control C (CTRLC)
- Average Control (AVGCTRL)
- Sampling Time Control (SAMPCTRL)
- Window Monitor Lower Threshold (WINLT)
- Window Monitor Upper Threshold (WINUT)
- Gain Correction (GAINCORR)
- Offset Correction (OFFSETCORR)

When one of these registers is written, the data is stored in the corresponding buffer as long as the current conversion is not impacted, and the corresponding busy status will be set in the Synchronization Busy register (SYNCBUSY). When a new RESULT is available, data stored in the buffer registers will be transferred to the ADC, and a new conversion can start.

13.27.6.3.2 Device Temperature Measurement

Principle
The device has an integrated temperature sensor, which is part of the Supply Controller (SUPC). The analog signal of that sensor can be converted into a digital value by the ADC. The digital value can be converted into a temperature in °C by following the steps in this section.

Configuration and Conditions
In order to conduct temperature measurements, configure the device according to these steps.

1. Configure the clocks and device frequencies according to the Electrical Characteristics.
2. Configure the Voltage References System of the Supply Controller (SUPC):
   2.1. Enable the temperature sensor by writing a '1' to the Temperature Sensor Enable bit in the VREF Control register (SUPC.VREF.TSEN).
   2.2. Select the required voltage for the internal voltage reference INTREF by writing to the Voltage Reference Selection bits (SUPC.VREF.SEL). The required value can be found in the Electrical Characteristics.
3. Configure the ADC:
   3.1. Select the internal voltage reference INTREF as ADC reference voltage by writing to the Reference Control register (ADC.REFCTRL.REFSEL).
   3.2. Select the temperature sensor vs. internal GND as input by writing TEMP and GND to the positive and negative MUX Input Selection bit fields (ADC.INPUTCTRL.MUXNEG and .MUXPOS, respectively).
   3.3. Configure the remaining ADC parameters according to the Electrical Characteristics.
   3.4. Enable the ADC and acquire a value, ADC_m.

Calculation Parameter Values
The temperature sensor behavior is linear, but it is sensitive to several parameters such as the internal voltage reference, which, itself, depends on the temperature. To take this into account, each device contains a Temperature Log row with individual calibration data measured and written during the production tests. These calibration values are read by software to provide the most accurate temperature readings possible.

The Temperature Log Row contains the following parameter set for two different temperatures ("ROOM" and "HOT"):

- Calibration temperatures in °C. One at room temperature $temp_R$, one at a higher temperature $temp_H$:
  - ROOM_TEMP_VAL_INT and ROOM_TEMP_VAL_DEC contain the measured temperature at room insertion, $temp_R$, in °C, separated in integer and decimal value.
    Example: For ROOM_TEMP_VAL_INT = 0x19 = 25 and ROOM_TEMP_VAL_DEC = 2, the measured temperature at room insertion is 25.2°C.
  - HOT_TEMP_VAL_INT and HOT_TEMP_VAL_DEC contain the measured temperature at hot insertion, $temp_H$, in °C. The integer and decimal value are also separated.
- For each temperature, the corresponding sensor value at the ADC in 12-bit, ADC_R and ADC_H:
  - ROOM_ADC_VAL contains the 12-bit ADC value, ADC_R, corresponding to $temp_R$. Its conversion to Volt is denoted V_{ADCR}.
  - HOT_ADC_VAL contains the 12-bit ADC value, ADC_H, corresponding to $temp_H$. Its conversion to Volt is denoted V_{ADCH}.
- Actual reference voltages at each calibration temperature in Volt, $INT1V_R$ and $INT1V_H$, respectively:
  - ROOM_INT1V_VAL is the 2's complement of the internal 1V reference value at $temp_R$: $INT1V_R$.
  - HOT_INT1V_VAL is the 2's complement of the internal 1V reference value at $temp_H$: $INT1V_H$.
  - Both ROOM_INT1V_VAL and HOT_INT1V_VAL values are centered around 1V with a 0.001V step. In other words, the range of values [0,127] corresponds to [1V, 0.873V] and the range of values [-1, -127] corresponds to [1.001V, 1.127V]. INT1V == 1 – (VAL/1000) is valid for both ranges.

Calculating the Temperature by Linear Interpolation
Using the data pairs ($temp_R$, V_{ADCR}) and ($temp_H$, V_{ADCH}) for a linear interpolation, we have the following equation:

$$(\frac{V_{ADC} - V_{ADCR}}{temp - temp_R}) = (\frac{V_{ADCH} - V_{ADCR}}{temp_H - temp_R})$$

The voltages V_x are acquired as 12-bit ADC values ADC_x, with respect to an internal reference voltage $INT1V_x$:

[Equation 1]

$$V_{ADCx} = ADC_x \cdot \frac{\mathrm{INT1V}_x}{2^{12} - 1}$$

For the measured value of the temperature sensor, ADC_m, the reference voltage is assumed to be perfect, i.e., $INT1V_m = INT1V_c = 1V$. These substitutions yield a coarse value of the measured temperature $temp_C$:

[Equation 2]

$$temp_C = temp_R + \left[\frac{\left\{ \left(ADC_m \cdot \frac{\mathrm{INT1V}_c}{\left(2^{12} - 1\right)} \right) - \left(ADC_R \cdot \frac{\mathrm{INT1V}_R}{\left(2^{12} - 1\right)} \right) \right\} \cdot (temp_H - temp_R)}{\left(ADC_H \cdot \frac{\mathrm{INT1V}_H}{\left(2^{12} - 1\right)} \right) - \left(ADC_R \cdot \frac{\mathrm{INT1V}_R}{\left(2^{12} - 1\right)} \right)} \right]$$

Or, after eliminating the 12-bit scaling factor (2^{12}-1):

[Equation 3]

$$temp_C = temp_R + \left[\frac{\{ADC_m \cdot \mathrm{INT1V}_c - (ADC_R \cdot \mathrm{INT1V}_R)\} \cdot (temp_H - temp_R)}{\{(ADC_H \cdot \mathrm{INT1V}_H) - (ADC_R \cdot \mathrm{INT1V}_R)\}} \right]$$

Equations 3 is a coarse value, because we assumed that $INT1V_c = 1V$. To achieve a more accurate result, we replace $INT1V_c$ with an interpolated value $INT1V_m$. We use the two data pairs ($temp_R$, $INT1V_R$) and ($temp_H$, $INT1V_H$) and yield:

$$(\frac{\mathrm{INT1V}_m - \mathrm{INT1V}_R}{temp_m - temp_R}) = (\frac{\mathrm{INT1V}_H - \mathrm{INT1V}V_R}{temp_H - temp_R})$$

Using the coarse temperature value $temp_c$, we can infer a more precise $INT1V_m$ value during the ADC conversion as:

[Equation 4]

$$\mathrm{INT1V}_m = \mathrm{INT1V}_R + (\frac{(\mathrm{INT1V}_H - \mathrm{INT1V}_R) \cdot (temp_C - temp_R)}{(temp_H - temp_R)})$$

Back to Equation 3, we replace the simple $INT1V_c = 1V$ by the more precise $INT1V_m$ of Equation 4, and find a more accurate temperature value $temp_f$:

[Equation 5]

$$temp_f = temp_R + \left[\frac{\{ADC_m \cdot \mathrm{INT1V}_m - (ADC_R \cdot \mathrm{INT1V}_R)\} \cdot (temp_H - temp_R)}{\{(ADC_H \cdot \mathrm{INT1V}_H) - (ADC_R \cdot \mathrm{INT1V}_R)\}} \right]$$

13.27.6.4 DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): the request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.

13.27.6.5 Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

The RESRDY and WINMON interrupts are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing

a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the ADC is reset. See 13.27.8.7 INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on the system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

13.27.6.6 Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to 13.27.8.4 EVCTRL for details.
- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to 13.27.8.10 CTRLC for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to 13.27.8.17 SWTRIG for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to 13.27.8.17 SWTRIG for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The ADC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

Related Links

13.18 EVSYS – Event System

13.27.6.7 Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 13-96.

Note: When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

Table 13-96. ADC Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
x	x	0	Disabled
0	0	1	Run in all sleep modes except STANDBY.
0	1	1	Run in all sleep modes on request, except STANDBY.
1	0	1	Run in all sleep modes.
1	1	1	Run in all sleep modes on request.

13.27.6.8 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- Input Control register (INPUTCTRL)
- Control C register (CTRLC)
- Average control register (AVGCTRL)
- Sampling time control register (SAMPCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)
- Gain correction register (GAINCORR)
- Offset Correction register (OFFSETCORR)
- Software Trigger register (SWTRIG)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

13.4.3 Register Synchronization

13.27.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0						PRESCALER[2:0]		
0x02	REFCTRL	7:0	REFCOMP				REFSEL[3:0]			
0x03	EVCTRL	7:0			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
0x04	INTENCLR	7:0						WINMON	OVERRUN	RESRDY
0x05	INTENSET	7:0						WINMON	OVERRUN	RESRDY
0x06	INTFLAG	7:0						WINMON	OVERRUN	RESRDY
0x07	SEQSTATUS	7:0	SEQBUSY			SEQSTATE[4:0]				
0x08	INPUTCTRL	7:0				MUXPOS[4:0]				
		15:8				MUXNEG[4:0]				
0x0A	CTRLC	7:0			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
		15:8						WINMODE[2:0]		
0x0C	AVGCTRL	7:0		ADJRES[2:0]			SAMPLENUM[3:0]			
0x0D	SAMPCTRL	7:0	OFFCOMP		SAMPLEN[5:0]					
0x0E	WINLT	7:0	WINLT[7:0]							
		15:8	WINLT[15:8]							
0x10	WINUT	7:0	WINUT[7:0]							
		15:8	WINUT[15:8]							
0x12	GAINCORR	7:0	GAINCORR[7:0]							
		15:8					GAINCORR[11:8]			
0x14	OFFSETCORR	7:0	OFFSETCORR[7:0]							
		15:8					OFFSETCORR[11:8]			
0x16 ... 0x17	Reserved									
0x18	SWTRIG	7:0							START	FLUSH
0x19 ... 0x1B	Reserved									
0x1C	DBGCTRL	7:0								DBGRUN
0x1D ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
		15:8						SWTRIG	OFFSETCORR	GAINCORR
0x22 ... 0x23	Reserved									
0x24	RESULT	7:0	RESULT[7:0]							
		15:8	RESULT[15:8]							
0x26 ... 0x27	Reserved									
0x28	SEQCTRL	7:0	SEQENn[7:0]							
		15:8	SEQENn[15:8]							
		23:16	SEQENn[23:16]							
		31:24	SEQENn[31:24]							
0x2C	CALIB	7:0						BIASCOMP[2:0]		
		15:8						BIASREFBUF[2:0]		

13.27.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 13.27.5.8 Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to 13.27.6.8 Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.27.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	SWRST
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests. In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.
If On Demand is disabled the ADC will always be running when enabled.
In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.
This bit is not synchronized.

Value	Description
0	The ADC is always on , if enabled.
1	The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the ADC behaves during standby sleep mode.
This bit is not synchronized.

Value	Description
0	The ADC is halted during standby sleep mode.
1	The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode.

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The ADC is disabled.
1	The ADC is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.
Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.27.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						PRESCALER[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – PRESCALER[2:0] Prescaler Configuration
This field defines the ADC clock relative to the peripheral clock.

Value	Name	Description
0x0	DIV2	Peripheral clock divided by 2
0x1	DIV4	Peripheral clock divided by 4
0x2	DIV8	Peripheral clock divided by 8
0x3	DIV16	Peripheral clock divided by 16
0x4	DIV32	Peripheral clock divided by 32
0x5	DIV64	Peripheral clock divided by 64
0x6	DIV128	Peripheral clock divided by 128
0x7	DIV256	Peripheral clock divided by 256

13.27.8.3 Reference Control

Name: REFCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	REFCOMP				REFSEL[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – REFCOMP Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will decrease the input impedance and, thus, increase the start-up time of the reference.

Value	Description
0	Reference buffer offset compensation is disabled.
1	Reference buffer offset compensation is enabled.

Bits 3:0 – REFSEL[3:0] Reference Selection
These bits select the reference for the ADC.

Value	Name	Description
0x0	INTREF	Internal variable reference voltage
x01	INTVCC0	1/1.6 VDDANA
0x2	INTVCC1	1/2 VDDANA (only for VDDANA > 2.0V)
0x3	—	—
0x4	VREFB	External reference
0x5	INTVCC2	VDDANA
0x6 - 0xF	—	Reserved

13.27.8.4 Event Control

Name: EVCTRL
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – WINMONEO Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is enabled and an event will be generated.

Bit 4 – RESRDYEO Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Value	Description
0	Result Ready event output is disabled and an event will not be generated.
1	Result Ready event output is enabled and an event will be generated.

Bit 3 – STARTINV Start Conversion Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 2 – FLUSHINV Flush Event Invert Enable

Value	Description
0	Flush event input source is not inverted.
1	Flush event input source is inverted.

Bit 1 – STARTEI Start Conversion Event Input Enable

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – FLUSHEI Flush Event Input Enable

Value	Description
0	A flush and new conversion will not be triggered on any incoming event.
1	A flush and new conversion will be triggered on any incoming event.

13.27.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The window monitor interrupt is disabled.
1	The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set.

13.27.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled.

13.27.8.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN Overrun
This flag is cleared by writing a '1' to the flag.
This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENCLR/SET.OVERRUN=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY Result Ready
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Result Ready interrupt flag.

13.27.8.8 Sequence Status

Name: SEQSTATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	SEQBUSY			SEQSTATE[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – SEQBUSY Sequence busy
This bit is set when the sequence start.
This bit is clear when the last conversion in a sequence is done.

Bits 4:0 – SEQSTATE[4:0] Sequence State
These bit fields are the pointer of sequence. This value identifies the last conversion done in the sequence.

13.27.8.9 Input Control

Name: INPUTCTRL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
				MUXNEG[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				MUXPOS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – MUXNEG[4:0] Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x18	GND	Internal ground
0x19 – 0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0] Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage or temperature sensor input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A	AIN10	ADC AIN10 pin
0x0B	AIN11	ADC AIN11 pin
0x0C	AIN12	ADC AIN12 pin
0x0D	AIN13	ADC AIN13 pin
0x0E	AIN14	ADC AIN14 pin
0x0F	AIN15	ADC AIN15 pin
0x10	AIN16	ADC AIN16 pin
0x11	AIN17	ADC AIN17 pin
0x12	AIN18	ADC AIN18 pin
0x13	AIN19	ADC AIN19 pin

Value	Name	Description
0x14 - 0x17	-	Reserved
0x18	TEMP	Temperature Sensor
0x19	BANDGAP	Bandgap Voltage
0x1A	SCALEDCOREVCC	1/4 Scaled Core Supply
0x1B	SCALEDIOVCC	1/4 Scaled I/O Supply
0x1D	SCALEDVBAT	1/4 Scaled VBAT Supply
0x1E	TEMP (CTAT)	Temperature Sensor

13.27.8.10 Control C

Name: CTRLC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						WINMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 10:8 – WINMODE[2:0] Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT
0x4	MODE4	WINUT < RESULT < WINLT
0x5 - 0x7		Reserved

Bits 5:4 – RESSEL[1:0] Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	For averaging mode output
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN Free Running Mode

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

13.27.8.11 Average Control

Name: AVGCTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:4 – ADJRES[2:0] Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0] Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples
0x2	4 samples
0x3	8 samples
0x4	16 samples
0x5	32 samples
0x6	64 samples
0x7	128 samples
0x8	256 samples
0x9	512 samples
0xA	1024 samples
0xB - 0xF	Reserved

13.27.8.12 Sampling Time Control

Name: SAMPCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	OFFCOMP		SAMPLEN[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – OFFCOMP Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. This compensation increases the sampling time by three clock cycles.
This bit must be set to zero to validate the SAMPLEN value. It's not possible to use OFFCOMP=1 and SAMPLEN>0.

Bits 5:0 – SAMPLEN[5:0] Sampling Time Length
These bits control the ADC sampling time in number of CLK_ADC cycles, depending of the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:
$\text{Sampling time} = (\text{SAMPLEN}+1) \cdot (\text{CLK}_{\text{ADC}})$

13.27.8.13 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINLT[15:0] Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

13.27.8.14 Window Monitor Upper Threshold

Name: WINUT
Offset: 0x10
Reset: 0x0000
Property: PAV Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINUT[15:0] Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

13.27.8.15 Gain Correction

Name: GAINCORR
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					GAINCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – GAINCORR[11:0] Gain Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

13.27.8.16 Offset Correction

Name: OFFSETCORR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					OFFSETCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – OFFSETCORR[11:0] Offset Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.

13.27.8.17 Software Trigger

Name: SWTRIG
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							START	FLUSH
Access							W	W
Reset							0	0

Bit 1 – START ADC Start Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.
Writing a '0' to this bit will have no effect.

Bit 0 – FLUSH ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit is cleared until the ADC has been flushed.
After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.
Writing this bit to '0' will have no effect.

13.27.8.18 Debug Control

Name: DBGCTRL
Offset: 0x1C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.
This bit should be written only while a conversion is not ongoing.

Value	Description
0	The ADC is halted when the CPU is halted by an external debugger.
1	The ADC continues normal operation when the CPU is halted by an external debugger.

13.27.8.19 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						SWTRIG	OFFSETCORR	GAINCORR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SWTRIG Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.
This bit is set when the synchronization of SWTRIG register between clock domains is started.

Bit 9 – OFFSETCORR Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.
This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

Bit 8 – GAINCORR Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.
This bit is set when the synchronization of GAINCORR register between clock domains is started.

Bit 7 – WINUT Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.
This bit is set when the synchronization of WINUT register between clock domains is started.

Bit 6 – WINLT Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.
This bit is set when the synchronization of WINLT register between clock domains is started.

Bit 5 – SAMPCTRL Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.
This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

Bit 4 – AVGCTRL Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.
This bit is set when the synchronization of AVGCTRL register between clock domains is started.

Bit 3 – CTRLC Control C Synchronization Busy
This bit is cleared when the synchronization of CTRLC register between the clock domains is complete.
This bit is set when the synchronization of CTRLC register between clock domains is started.

Bit 2 – INPUTCTRL Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.
This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started

13.27.8.20 Result

Name: RESULT
Offset: 0x24
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	RESULT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESULT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RESULT[15:0] Result Conversion Value
These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.
In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLC.LEFTADJ.
If the result is left-adjusted (CTRLC.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.
If the result is not left-adjusted (CTRLC.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.

13.27.8.21 Sequence Control

Name: SEQCTRL
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SEQENn[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEQENn[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SEQENn[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEQENn[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEQENn[31:0] Enable Positive Input in the Sequence
For details on available positive mux selection, refer to INPUTCTRL.MUXENG.
The sequence start from the lowest input, and go to the next enabled input automatically when the conversion is done. If no bits are set the sequence is disabled.

Value	Description
0	Disable the positive input mux n selection from the sequence.
1	Enable the positive input mux n selection to the sequence.

13.27.8.22 Calibration

Name: CALIB
Offset: 0x2C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
						BIASREFBUF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						BIASCOMP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – BIASREFBUF[2:0] Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.
The value must be copied only, and must not be changed.

Bits 2:0 – BIASCOMP[2:0] Bias Comparator Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.
The value must be copied only, and must not be changed
Related Links
10.4 NVM Software Calibration Area Mapping

13.28 AC – Analog Comparators

13.28.1 Overview

The Analog Comparator (AC) supports two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis and propagation delay are two important properties of the comparators' dynamic behavior. Both parameters may be adjusted to achieve the optimal operation for each application.

The input selection includes four shared analog port pins and several internal signals. Each comparator output state can also be output on a pin for use by external devices.

The comparators are always grouped in pairs on each port. The AC peripheral implements one pair of comparators. These are called Comparator 0 (COMP0) and Comparator 1 (COMP1). They have identical behaviors, but separate control registers. The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

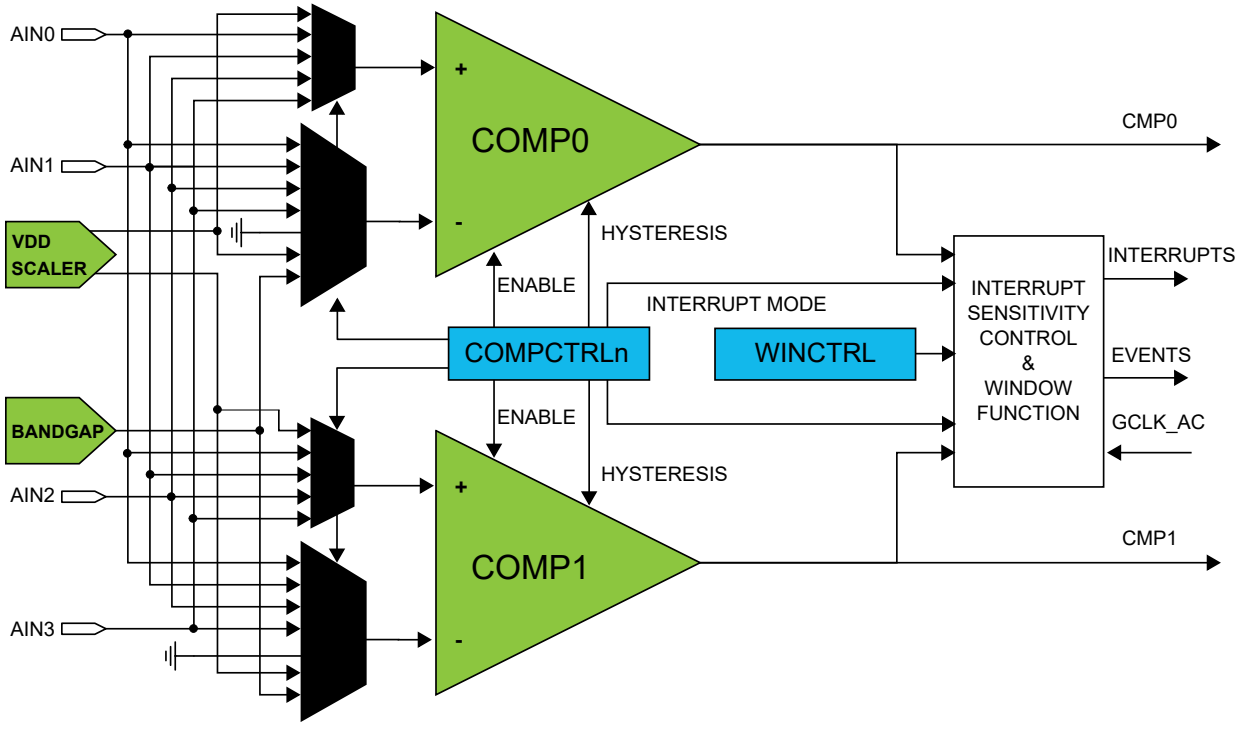
13.28.2 Features

- Two individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
  - 4-levels or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:

  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable VDD scaler per comparator
- Interrupt generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window function interrupt generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event generation on:
  - Comparator output
  - Window function inside/outside window
- Optional digital filter on comparator output
- Low-power option
  - Single-shot support

13.28.3 Block Diagram

Figure 13-211. Analog Comparator Block Diagram



13.28.4 Signal Description

Signal	Description	Type
AIN[3..0]	Analog input	Comparator inputs

.....continued		
Signal	**Description**	**Type**
CMP[1..0]	Digital output	Comparator outputs

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.28.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.28.5.1 I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

13.17 PORT - I/O Pin Controller

13.28.5.2 Power Management

The AC will continue to operate in any sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

13.8 PM – Power Manager

13.28.5.3 Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_AC_APB can be found in the Peripheral Clock Masking section in the Power Manager description.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

13.8 PM – Power Manager

13.28.5.4 DMA

Not applicable.

13.28.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

11.2 Nested Vector Interrupt Controller

13.28.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

13.18 EVSYS – Event System

13.28.5.7 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any ongoing comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to 13.28.8.9 DBGCTRL

for details. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

13.28.5.8 Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Control B register (CTRLB)
- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.28.5.9 Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. Each pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, must be configured and enabled prior to its use as a comparator input.

13.28.6 Functional Description

13.28.6.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (normal mode) or paired to form a window comparison (window mode).

13.28.6.2 Basic Operation

13.28.6.2.1 Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

13.28.6.2.2 Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled writing a '0' to CTRLA.ENABLE.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC will be reset to their initial state, and the AC will be disabled. Refer to *CTRLA* for details.

13.28.6.2.3 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLx.HYSTEN and COMPCTRLx.HYST. See Input Hysteresis for more details.
- Select the comparator speed versus power with COMPCTRLx.SPEED. See Propagation Delay vs. Power Consumption for more details.
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See Selecting Comparator Inputs for more details.

- Select the filtering option with COMPCTRLx.FLEN.
- Select standby operation with Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

13.28.6.2.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in *Electrical Characteristics*. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the single-shot mode to chain further events in the system, regardless of the state of the comparator outputs. The interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

Related Links

15. Electrical Characteristics

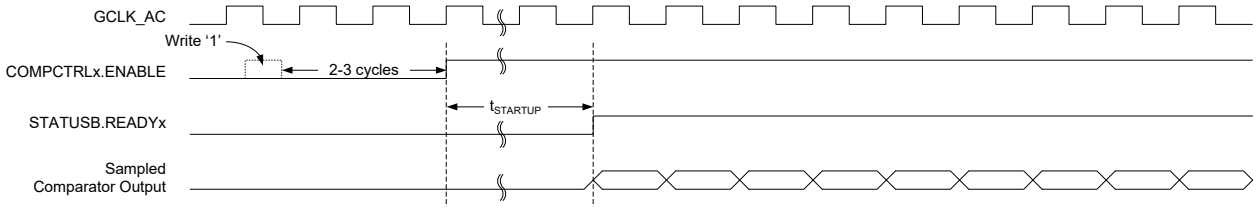
Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the Figure 13-212.

Figure 13-212. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

15. Electrical Characteristics

Single-Shot

Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has

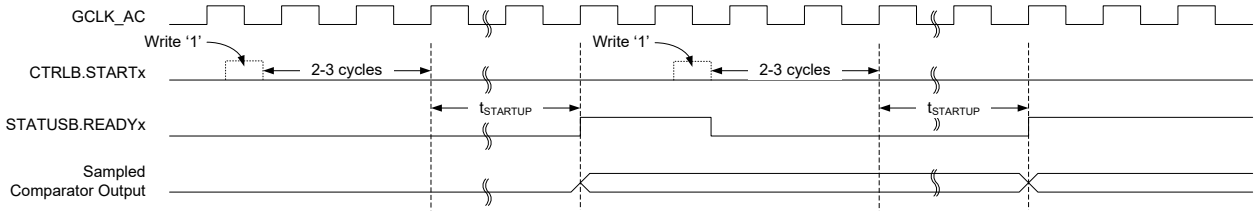
passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in Figure 13-213.

Figure 13-213. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

15. Electrical Characteristics

13.28.6.3 Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

13.28.6.4 Window Operation

Each comparator pair can be configured to work together in window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

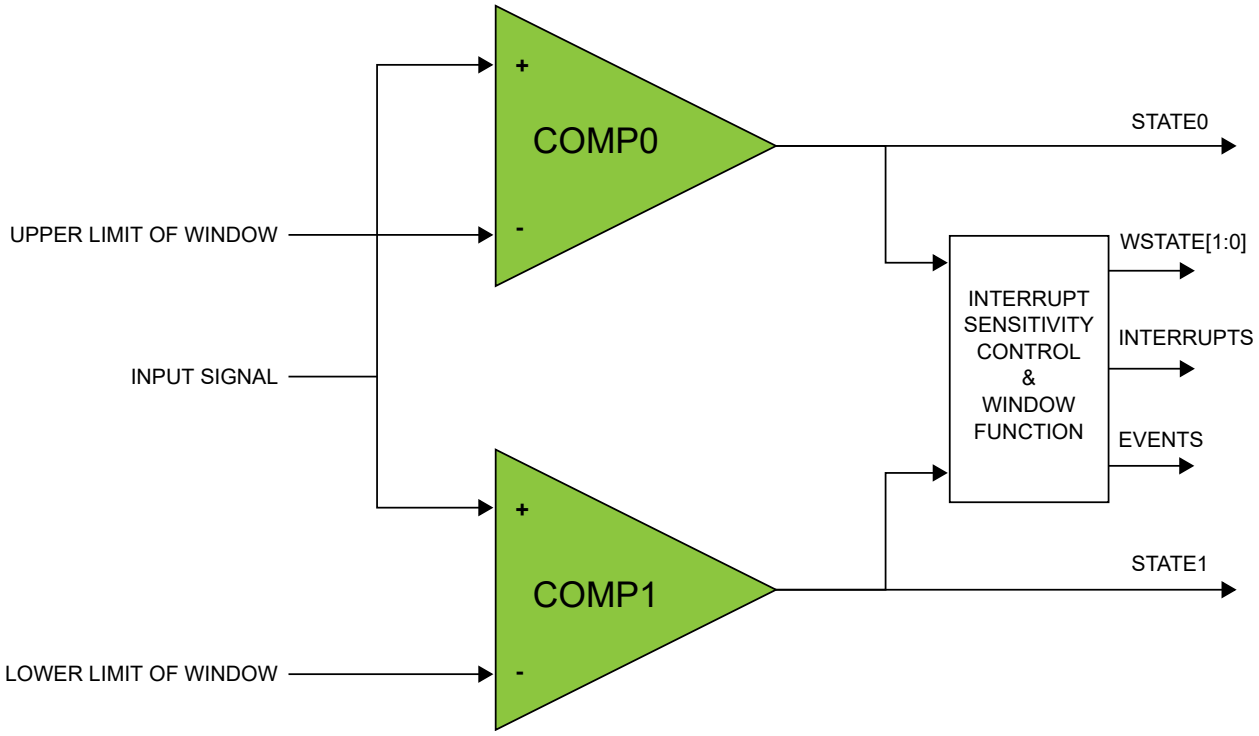
To physically configure the pair of comparators for window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In Figure 13-214, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input

voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during window mode.

When the comparators are configured for window mode and single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.

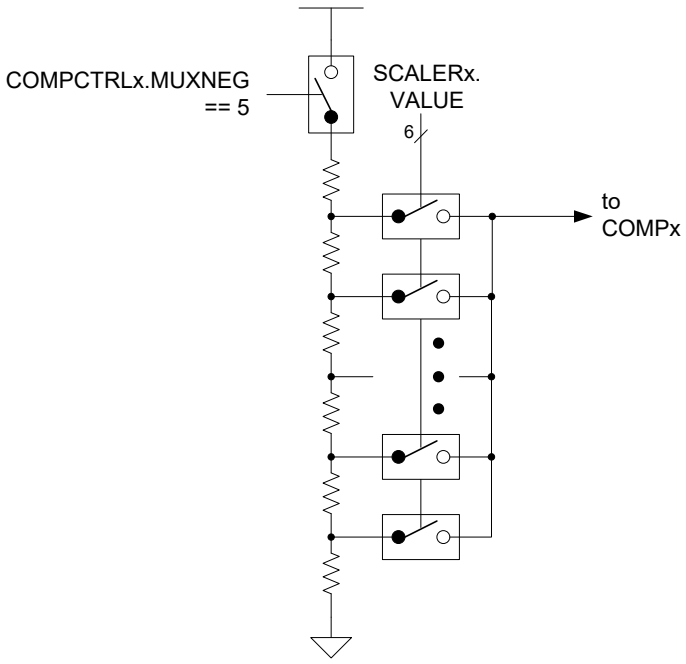
Figure 13-214. Comparators in Window Mode



13.28.6.5 VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scaler of a comparator is enabled when the Negative Input Mux bit field in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to 0x5 and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the Scaler x registers (SCALERx.VALUE).

Figure 13-215. VDD Scaler



13.28.6.6 Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST). Hysteresis is available only in continuous mode (COMPCTRLx.SINGLE=0).

13.28.6.7 Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

13.28.6.8 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 13-216. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 13-217.

Figure 13-216. Continuous Mode Filtering

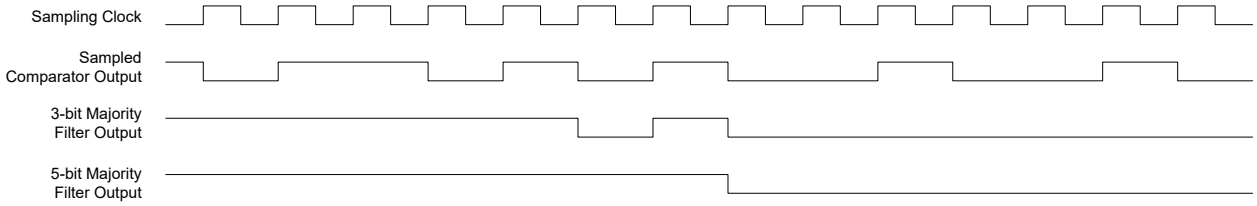
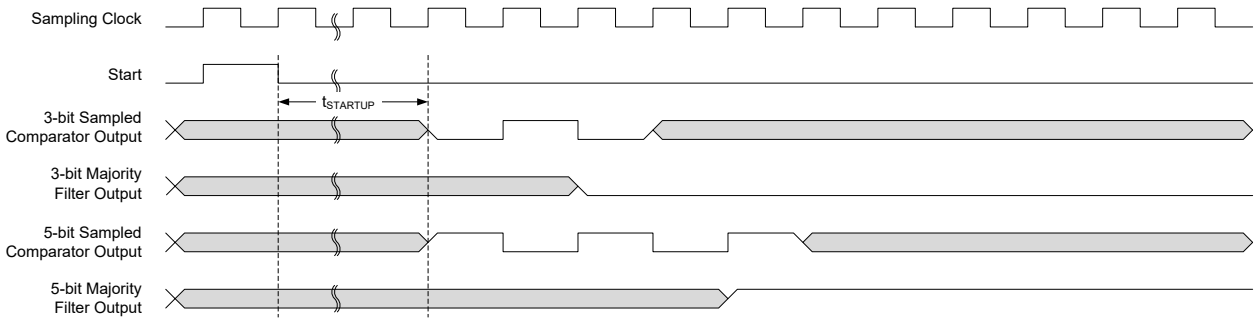


Figure 13-217. Single-Shot Filtering



During sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during sleep modes, or the resulting interrupt/event may be generated incorrectly.

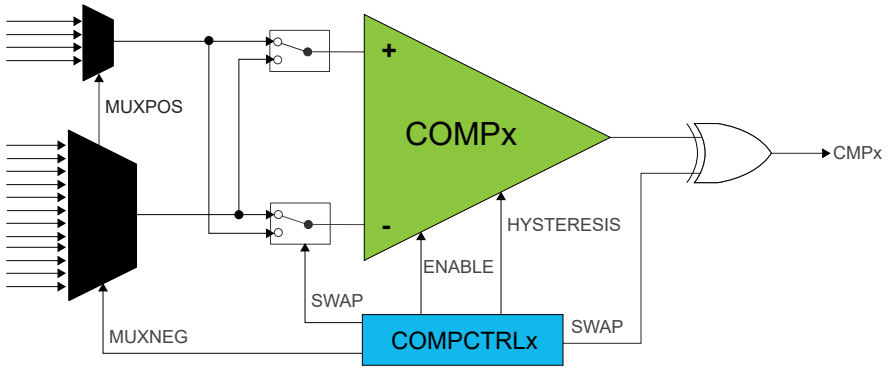
13.28.6.9 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

13.28.6.10 Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 13-218. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 13-218. Input Swapping for Offset Compensation



13.28.6.11 DMA Operation

Not applicable.

13.28.6.12 Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status.
- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRL.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL0).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until

the interrupt flag is cleared, the interrupt is disabled, or the AC is reset. See INFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

13.28.6.13 Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

13.28.6.14 Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 13-97.

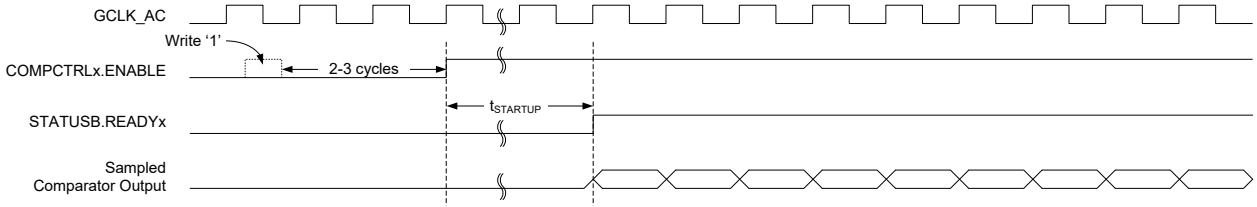
Table 13-97. Sleep Mode Operation

COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

13.28.6.14.1 Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

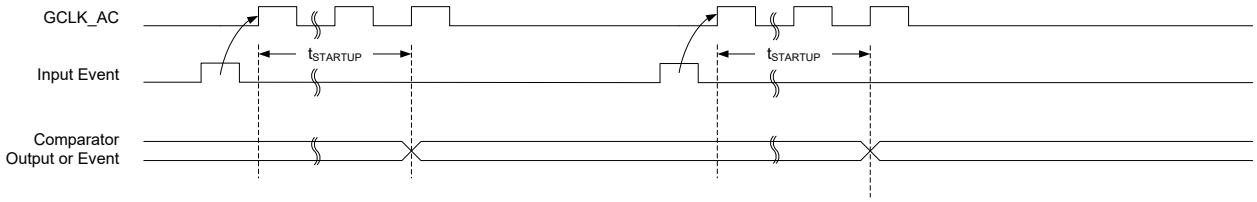
Figure 13-219. Continuous Mode SleepWalking



13.28.6.14.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 13-220. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 13-220. Single-Shot SleepWalking



13.28.6.15 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

13.4.3 Register Synchronization

13.28.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							START1	START0
0x02	EVCTRL	7:0				WINEO0			COMPEO1	COMPEO0
		15:8			INVEI1	INVEI0			COMPEI1	COMPEI0
0x04	INTENCLR	7:0				WIN0			COMP1	COMP0
0x05	INTENSET	7:0				WIN0			COMP1	COMP0
0x06	INTFLAG	7:0				WIN0			COMP1	COMP0
0x07	STATUSA	7:0			WSTATE0[1:0]				STATE1	STATE0
0x08	STATUSB	7:0							READY1	READY0
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0			VALUE[5:0]					
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		31:24			OUT[1:0]			FLEN[2:0]		
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		31:24			OUT[1:0]			FLEN[2:0]		
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
		15:8								
		23:16								
		31:24								

13.28.8 Register Description

Registers can be 8, 16 or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC – Peripheral Access Controller* and *Synchronization* for details.

13.28.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	W
Reset							0	0

Bit 1 – ENABLE Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

13.28.8.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – STARTx Comparator x Start Comparison
Writing a '0' to this field has no effect.
Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.
This bit always reads as zero.

13.28.8.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEI1	INVEI0			COMPEI1	COMPEI0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEO1	COMPEO0
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 12, 13 – INVEIx Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 8, 9 – COMPEIx Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.
These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 4 – WINEO0 Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 0, 1 – COMPEOx Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

13.28.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

13.28.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

13.28.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 0, 1 – COMPx Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.
This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator x interrupt flag.

13.28.8.7 Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATE1	STATE0
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

Bits 0, 1 – STATEx Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

13.28.8.8 Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
							READY1	READY0
Access							R	R
Reset							0	0

Bits 0, 1 – READYx Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.

13.28.8.9 Debug Control

Name: DBGCTRL
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

13.28.8.10 Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0] Window 0 Interrupt Selection
These bits configure the interrupt mode for the comparator window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0 Window 0 Mode Enable

Value	Description
0	Window mode is disabled for comparators 0 and 1.
1	Window mode is enabled for comparators 0 and 1.

13.28.8.11 Scaler n

Name: SCALER
Offset: 0x0C + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			VALUE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – VALUE[5:0] Scaler Value
These bits define the scaling factor for channel n of the V_{DD} voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{\mathrm{SCALE}} = \frac{V_{\mathrm{DD}} \cdot (\mathrm{VALUE}+1)}{64}$$

13.28.8.12 Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
			HYST[1:0]		HYSTEN		SPEED[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0] Output
These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.
Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0] Filter Length
These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bits 21:20 – HYST[1:0] Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN = 1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE = 0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	HYST50	50 mV

Value	Name	Description
0x1	HYST70	70 mV
0x2	HYST90	90 mV
0x3	HYST110	110 mV

Bit 19 – HYSTEN Hysteresis Enable
This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE = 0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.
This bit is not synchronized.

Value	Description
0	Hysteresis is disabled.
1	Hysteresis is enabled.

Bits 17:16 – SPEED[1:0] Speed Selection
This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	LOW	Low speed
0x1	MEDLOW	Medium low speed
0x2	MEDHIGH	Medium high speed
0x3	HIGH	High speed

Bit 15 – SWAP Swap Inputs and Invert
This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0] Positive Input Mux Selection
These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	VSCALE	VDD scaler
0x5-0x7	—	Reserved

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection
These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler

Value	Name	Description
0x6	BANDGAP	Internal bandgap voltage
0x7	Reserved	Reserved

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of the comparator during standby sleep mode.
This bit is not synchronized.

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0] Interrupt Selection
These bits select the condition for comparator n to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 2 – SINGLE Single-Shot Mode
This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE Enable
Writing a ‘0’ to this bit disables comparator n.
Writing a ‘1’ to this bit enables comparator n.
Due to synchronization, there is a delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.
Writing a ‘1’ to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to ‘0’ and the write is synchronized.

13.28.8.13 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 3, 4 – COMPCTRLx COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.
This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

Bit 2 – WINCTRL WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.
This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

13.29 PTC - Peripheral Touch Controller

13.29.1 Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect a touch on the capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self and mutual capacitance sensors.

In the Mutual Capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In the Self Capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

The number of available pins and the assignment of X- and Y-lines is depending on both package type and device configuration. Refer to the Configuration Summary and I/O Multiplexing table for details.

Related Links

7. I/O Multiplexing and Considerations

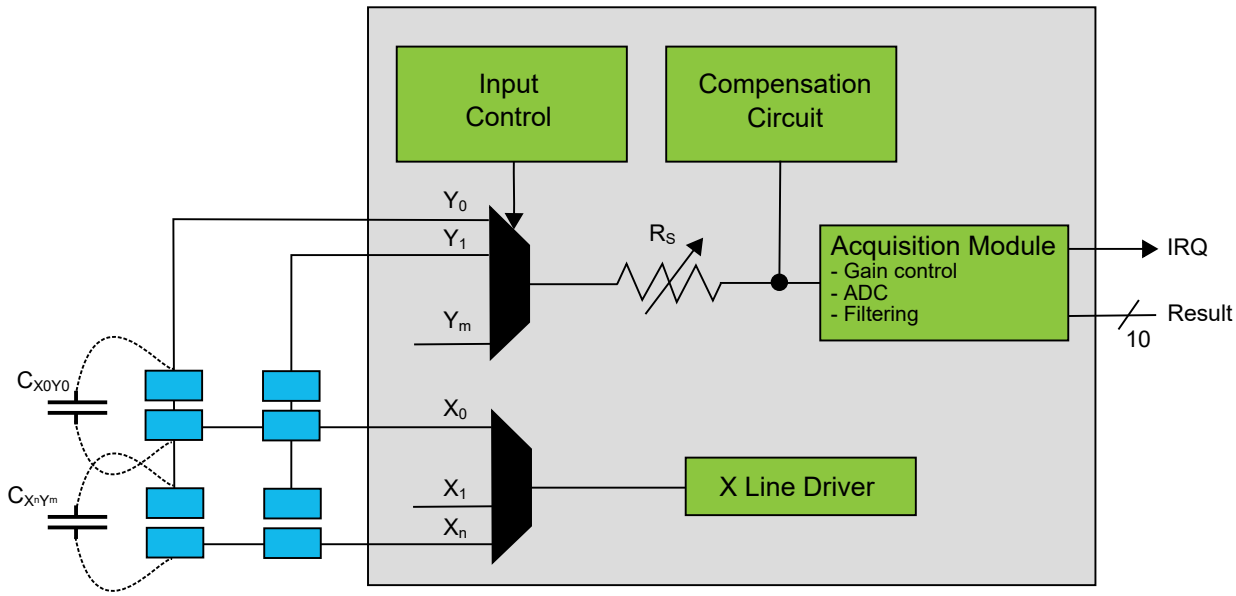
13.29.2 Features

- Low-Power, High-Sensitivity, Environmentally Robust Capacitive Touch Buttons, Sliders, Wheels and Proximity Sensing
  - Down to 8 µA with 200 ms scan rate
- Supports Wake-up on Touch from standby Sleep mode
- Supports Mutual Capacitance and Self Capacitance Sensing
  - 7/9 buttons in self-capacitance mode, for 32-/48- pins respectively
  - 12/18 buttons in mutual-capacitance mode, for 32-/48- pins respectively
  - Mix-and-match mutual and self capacitance sensors
- One Pin per Electrode – No External Components
- Load Compensating Charge Sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero Drift Over the Temperature and VDDANA Range
  - Auto calibration and recalibration of sensors
- Single-shot and free-running Charge Measurement
- Hardware Noise Filtering and Noise Signal Desynchronization for High Conducted Immunity
- Driven shield for better noise immunity and moisture tolerance
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete
  - 5% CPU utilization scanning 10 channels at 50 ms scan rate
- Using ADC peripheral for signal conversion and acquisition
- Supported by the Start QTouch® Configurator development tools, which comprises QTouch Library project builder and QTouch analyzer. See also Atmel|Start and Atmel Studio documentation.(1)

  Note:
  1. Support is directly available only for SAML21, which can be used for SAMR30.

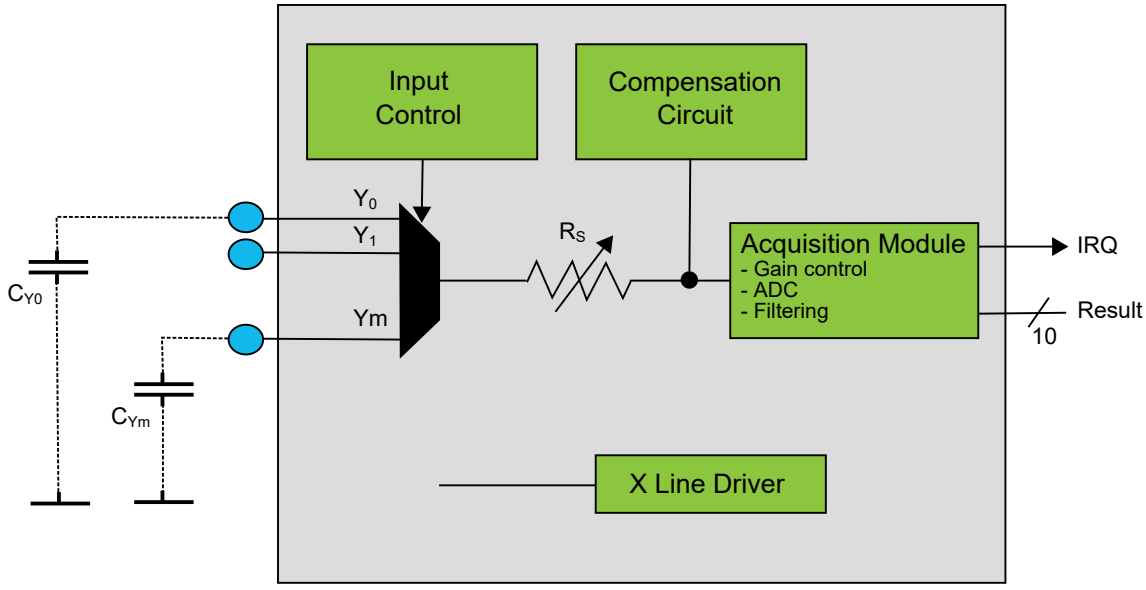
13.29.3 Block Diagram

Figure 13-221. PTC Block Diagram Mutual-Capacitance



Note: For SAM R30 the R_S = 20-100 KΩ.

Figure 13-222. PTC Block Diagram Self-Capacitance



Note: For SAM R30 the RS = 20-100 KΩ.

13.29.4 Signal Description

Table 13-98. Signal Description for PTC

Name	Type	Description
X[n:0]	Digital	X-line (Output)
Y[m:0]	Analog	Y-line (Input/Output)

Note: The number of X and Y lines are device dependent. Refer to *Configuration Summary* for details.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

7. I/O Multiplexing and Considerations

13.29.5 Product Dependencies

In order to use this Peripheral, configure the other components of the system as described in the following sections.

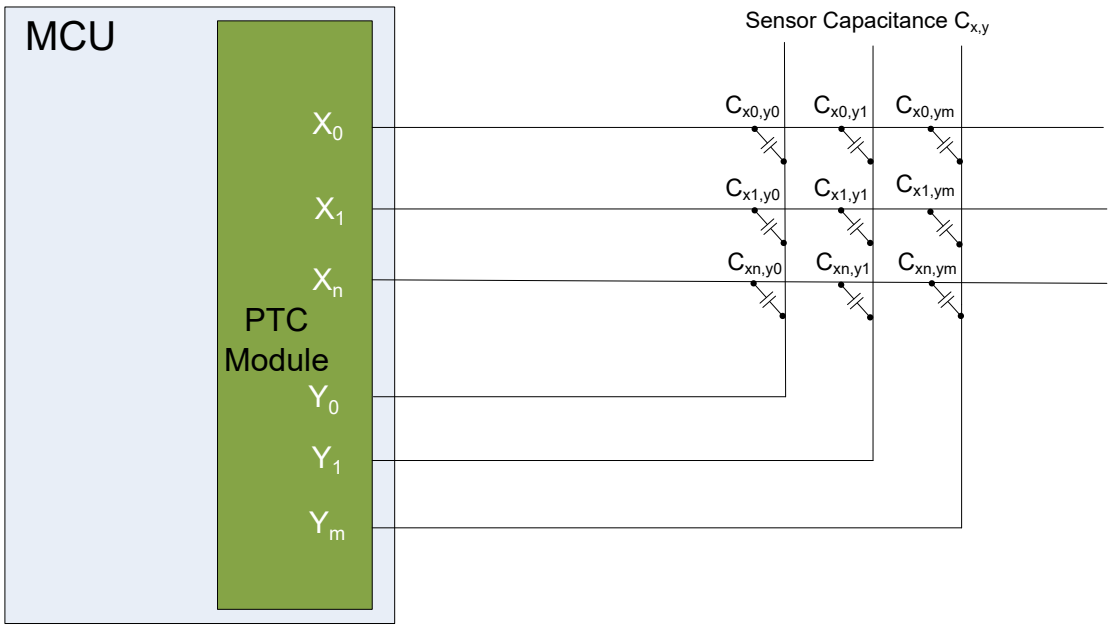
13.29.5.1 I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

13.29.5.1.1 Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

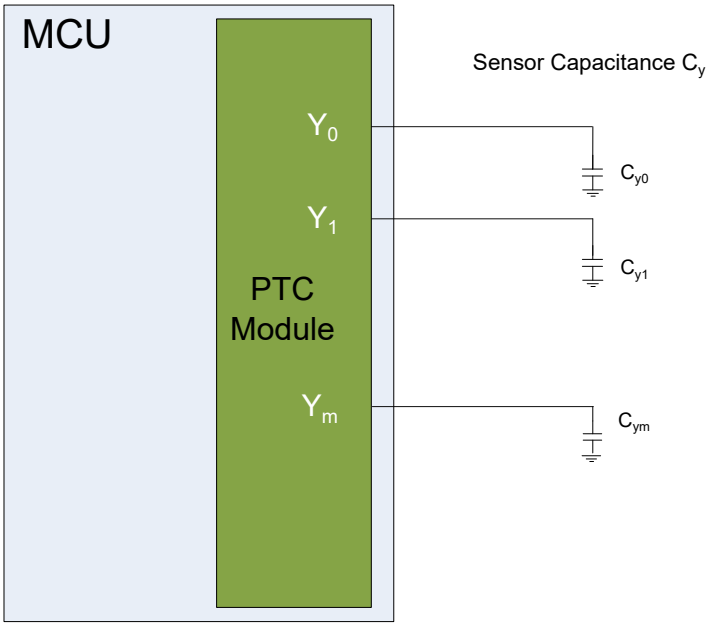
Figure 13-223. Mutual Capacitance Sensor Arrangement



13.29.5.1.2 Self-Capacitance Sensor Arrangement

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 13-224. Self-capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

13.29.5.2 Clocks

The PTC is clocked by the GCLK_PTC clock. CLK_PER clock. See the Related Links for details on configuring CLK_PER. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_APB). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller*.

The selected clock must be enabled in the Power Manager, before it can be used by the PTC. By default these clocks are disabled. The frequency range of GCLK_PTC is 400kHz to 4MHz.

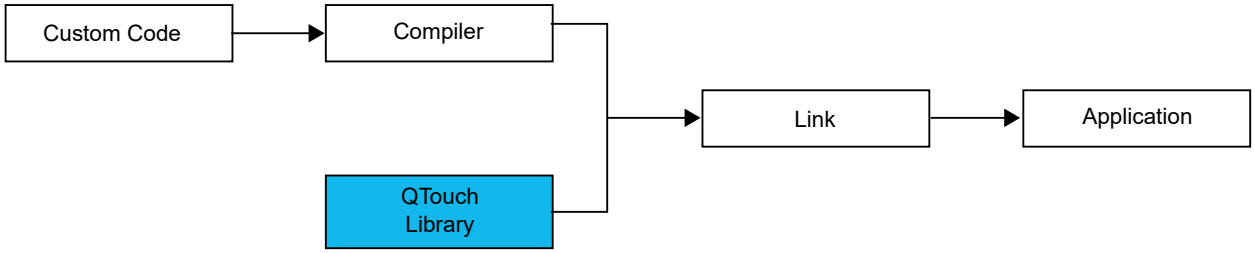
13.29.5.3 Analog-Digital Converter (ADC)

The PTC is using the ADC for signal conversion and acquisition. The ADC must be enabled and configured appropriatly in order to allow correct behavior of the PTC.

13.29.6 Functional Description

In order to access the PTC, the user must use the QTouch Composer tool to configure and link the QTouch Library firmware with the application software. QTouch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

Figure 13-225. QTouch Library Usage



For more information about QTouch Library, refer to the QTouch Library Peripheral Touch Controller User Guide.

13.30 RFCTRL – AT86RF212B Front-End Control Signal Interface

13.30.1 Overview

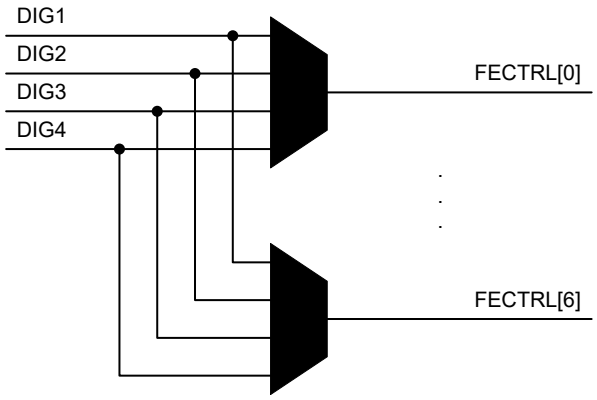
The RFCTRL module provides a register and multiplexer for selecting the front-end control signal outputs of the integrated transceiver as alternate pin functions for the SAM R30.

13.30.2 Features

- Supports up to 6 front-end control output signals
- Supports all front-end control input signals (DIG1, DIG2, DIG3 and DIG4) from the AT86RF212B.

13.30.3 Block Diagram

Figure 13-226. RFCTRL Block Diagram



13.30.4 Product Dependencies

In order to use the module, the I/O port pins used for front-end control signals must be configured as alternate pin function outputs.

13.30.5 Functional Description

The RFCTRL module is intended to flexibly route the PA/LNA and antenna diversity front-end control signals, as well as, the RX/TX frame time stamping to alternate pin functions of the Cortex-M0+ CPU without any further software action required after initialization of the alternate pin function output.

The module provides six 2-bit registers that control the muxing of the DIG1, DIG2, DIG3 and DIG4 front-end control signal outputs of the integrated transceiver to alternate pin functions of the SAM R30 device.

13.30.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	FECTRL	7:0	F3CFG[1:0]		F2CFG[1:0]		F1CFG[1:0]		F0CFG[1:0]	
		15:8					F5CFG[1:0]		F4CFG[1:0]	

13.30.7 Register Description

13.30.7.1 Front-End Control

Name: FECTRL
Offset: 0x00
Reset: 0x0000
Property: —

Bit	15	14	13	12	11	10	9	8
					F5CFG[1:0]		F4CFG[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	F3CFG[1:0]		F2CFG[1:0]		F1CFG[1:0]		F0CFG[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:10 – F5CFG[1:0]
These bits define the front-end output control signal 5, as shown in Table 13-99.

Bits 9:8 – F4CFG[1:0]
These bits define the front-end output control signal 4, as shown in Table 13-99.

Bits 7:6 – F3CFG[1:0]
These bits define the front-end output control signal 3, as shown in Table 13-99.

Bits 5:4 – F2CFG[1:0]
These bits define the front-end output control signal 2, as shown in Table 13-99.

Bits 3:2 – F1CFG[1:0]
These bits define the front-end output control signal 1, as shown in Table 13-99.

Bits 1:0 – F0CFG[1:0]
These bits define the front-end output control signal 0, as shown in Table 13-99.

Table 13-99. Front-End Control Configuration

FnCFG[1:0]	Operating Mode
0x0	Route transceiver DIG1 signal output to FECTRL[n] alternate pin function.
0x1	Route transceiver DIG2 signal output to FECTRL[n] alternate pin function.
0x2	Route transceiver DIG3 signal output to FECTRL[n] alternate pin function.
0x3	Route transceiver DIG4 signal output to FECTRL[n] alternate pin function.

14. Reference Guide - AT86RF212B

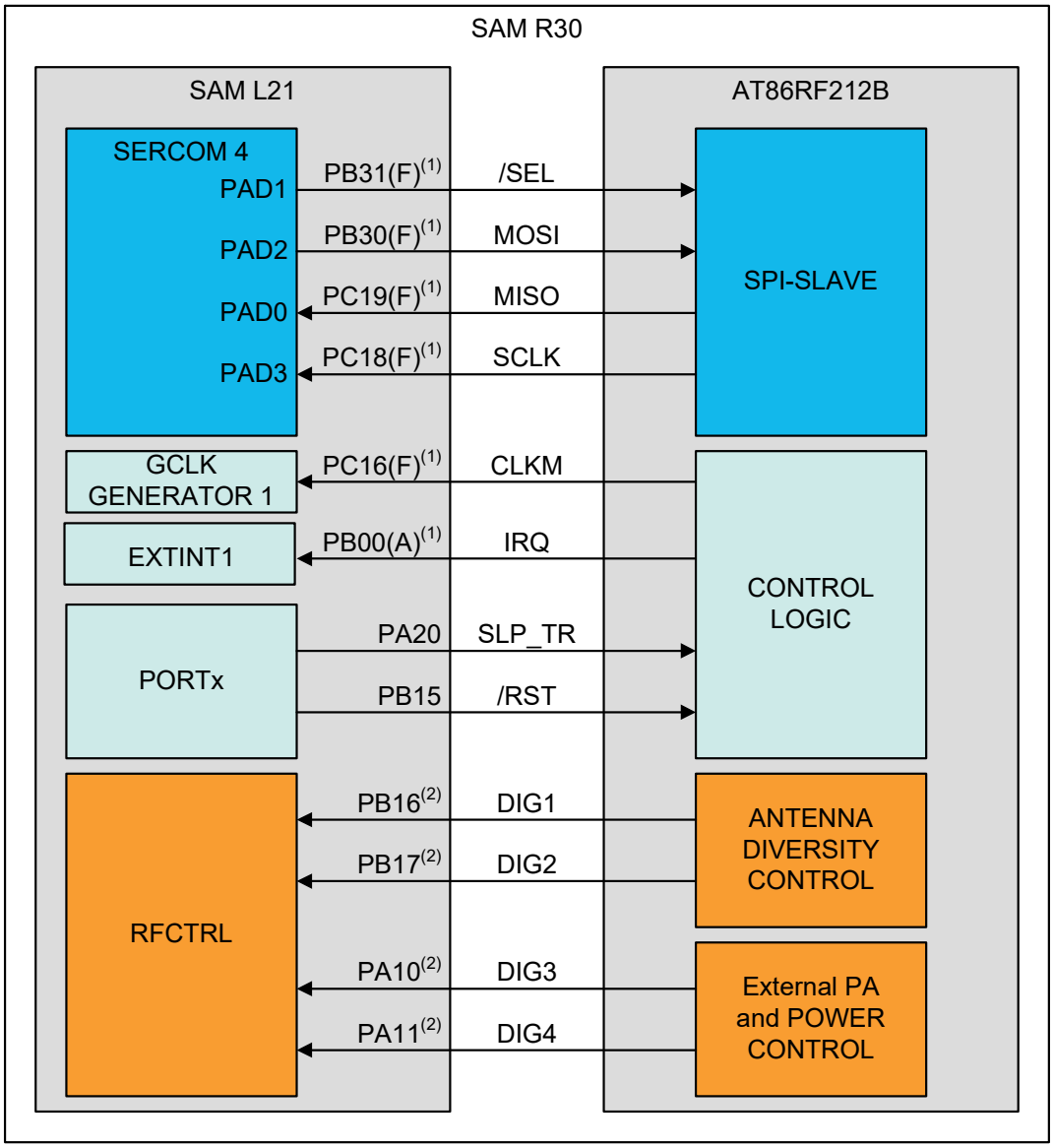
The SAM R30 system in package contains an AT86RF212B transceiver die. It is recommended to use available documentation, software sources and application notes for the AT86RF212B in addition.

14.1 Microcontroller Interface

14.1.1 Overview

This section describes the AT86RF212B to microcontroller interface. The interface comprises a slave SPI and additional control signals. This interface is connected to a SAM L21 master interface as shown below. The SPI timing and protocol are described in the sections below as well.

Figure 14-1. Microcontroller to AT86RF212B Interface



1. Alternate pin function and direction has to be configured by software.
2. Pin function is configured by hardware automatically after reset.

The SPI is used for register, Frame Buffer, SRAM, and AES access. The additional control signals are connected to the GPIO/IRQ interface of the microcontroller. The table below introduces the radio transceiver I/O signals and their functionality.

Table 14-1. Microcontroller Interface Signal Description

Signal	Description
/SEL	SPI select signal, active low
MOSI	SPI data (master output slave input) signal
MISO	SPI data (master input slave output) signal
SCLK	SPI clock signal

.....continued	
Signal	**Description**
CLKM	Optional, Clock output, usable as: - microcontroller clock source and/or MAC timer reference - high precision timing reference
IRQ	Interrupt request signal, further used as: - Frame Buffer Empty indicator
SLP_TR	Multi purpose control signal (functionality is state dependent): - Sleep/Wakeup: enable/disable SLEEP state - TX start: BUSY_TX_(ARET) state - disable/enable CLKM
/RST	AT86RF212B reset signal; active low
DIG2	Optional, - IRQ_2 (RX_START) for RX Frame Time Stamping

14.1.2 SPI Timing Description

CLKM can be used as a micro controller master clock source. If the micro controller generates the SPI master clock (SCLK) directly from CLKM, the SPI operates in synchronous mode, otherwise in asynchronous mode. Designing systems for synchronous mode, the wake up conditions need to be carefully checked to make sure the Transceiver wake up is triggered from al wake up event sources and the CLKM timing meets the controller requirements. For the SAM R30 SIP this mode of operation is not recommended.

In asynchronous mode, the maximum SCLK frequency fasync is limited to 7.5MHz. The signal CLKM is not required to derive SCLK and may be disabled to reduced power consumption and spurious emissions. Since CLKM is generated using a high accuracy crystal, it may be beneficial for many applications to run a timer from that clock.

The figures below illustrate the SPI timing and introduces its parameters. The corresponding timing parameter definitions t_1 – t_9 are defined in the *Digital Interface Timing Characteristics*.

Figure 14-2. SPI Timing, Global Map and Definition of Timing Parameters t_5, t_6, t_8, t_9.

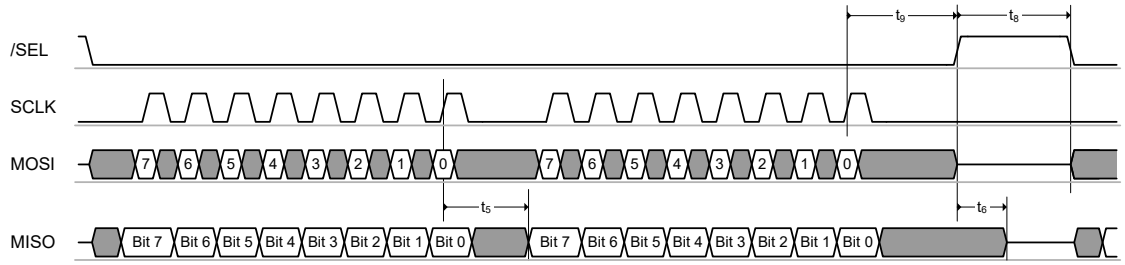
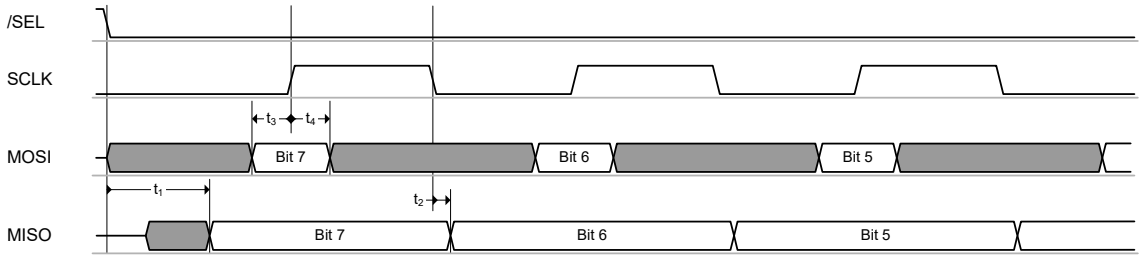


Figure 14-3. SPI Timing, Detailed Drawing of Timing Parameters t_1 to t_4.



The SPI is based on a byte-oriented protocol and is always a bidirectional communication between the master and slave. The SPI master starts the transfer by asserting /SEL = L. Then the master generates eight SPI clock cycles to

transfer one byte to the radio transceiver (via MOSI). At the same time, the slave transmits one byte to the master (via MISO). When the master wants to receive one byte of data from the slave, it must also transmit one byte to the slave. All bytes are transferred with the MSB first. An SPI transaction is finished by releasing /SEL = H.

An SPI register access consists of two bytes, a Frame Buffer or SRAM access of at least two or more bytes.

/SEL = L enables the MISO output driver of the AT86RF212B. The MSB of MISO is valid after t_1 and is updated on each SCLK falling edge. If the driver is disabled, there is no internal pull-up transistor connected to it. Driving the appropriate signal level must be ensured by the master device or an external pull-up resistor.

Note: When both /SEL and /RST are active, the MISO output driver is also enabled.

Referring to the figures above, AT86RF212B MOSI is sampled at the rising edge of the SCLK signal and the output is set at the falling edge of SCLK. The signal must be stable before and after the rising edge of SCLK as specified by t_3 and t_4.

This SPI operational mode is commonly known as "*SPI mode 0*".

Related Links

14.1.3 SPI Protocol

15.15.1 Digital Interface Timing Characteristics

14.1.3 SPI Protocol

Each SPI sequence starts with transferring a command byte from the SPI master via MOSI with the MSB first. This command byte defines the SPI access mode and additional mode-dependent information.

Table 14-2. SPI Command Byte Definition

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access Mode	Access Type
1	0	Register address [5:0]						Register access	Read access
1	1	Register address [5:0]							Write access
0	0	1	Reserved					Frame Buffer access	Read access
0	1	1	Reserved						Write access
0	0	0	Reserved					SRAM access	Read access
0	1	0	Reserved						Write access

Each SPI transfer returns bytes back to the SPI master on MISO output pin. The content of the first byte (see value *PHY_STATUS* in the following register-, frame buffer- and SRAM access mode figures) is set to zero after reset. To transfer status information of the radio transceiver to the microcontroller, the content of the first byte can be configured with the SPI_CMD_MODE bits in the TRX_CTRL_1 register (TRX_CTRL_1.SPI_CMD_MODE).

Note: Return values on MISO stated as XX shall be ignored by the microcontroller.

The different access modes are described within the following sections.

Related Links

14.8.4 TRX_CTRL_1

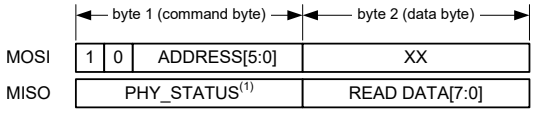
14.1.3.1 Register Access Mode

Register Access Mode is used to read and write AT86RF212B registers (register address from 0x00 up to 0x3F).

A register access mode is a two-byte read/write operation initiated by /SEL = L. The first transferred byte on MOSI is the command byte including an identifier bit (bit[7] = 1), a read/write select bit (bit[6]), and a 6-bit register address.

On read access, the content of the selected register address is returned in the second byte on MISO.

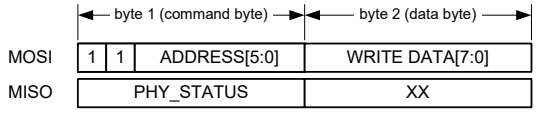
Figure 14-4. Packet Structure - Register Read Access



Note: Each SPI access can be configured to return radio controller status information (PHY_STATUS) on MISO.

On write access, the second byte transferred on MOSI contains the write data to the selected address.

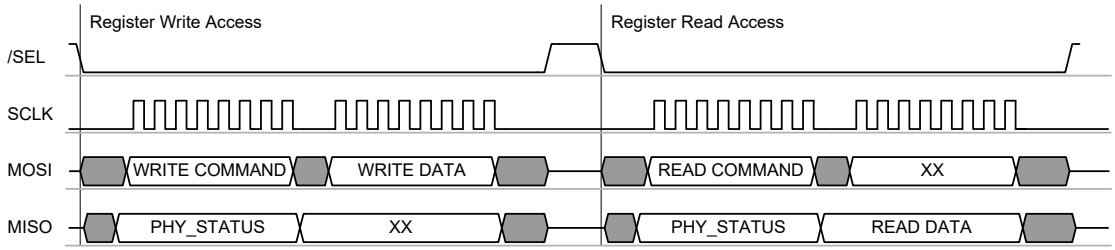
Figure 14-5. Packet Structure - Register Write Access



Each register access must be terminated by setting /SEL = H.

The figure below illustrates a typical SPI sequence for a register access sequence for write and read respectively.

Figure 14-6. Exemplary SPI Sequence – Register Access Mode



Related Links

14.1.4 Radio Transceiver Status Information

14.1.3.2 Frame Buffer Access Mode

Frame Buffer Access Mode is used to read and write AT86RF212B frame buffer. The frame buffer address is always reset to zero and incremented to access PSDU, LQI, ED and RX_STATUS data.

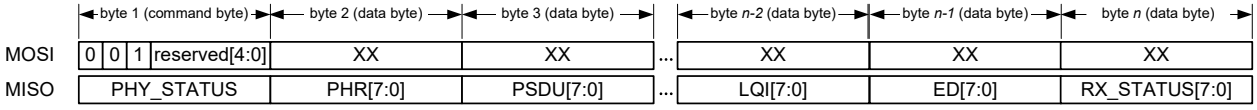
The Frame Buffer can hold up to 128-byte of one PHY service data unit (PSDU) IEEE 802.15.4 data frame. See related links for an introduction to the IEEE 802.15.4 frame format.

Each access starts with /SEL = L followed by a command byte on MOSI. Each frame read or write access command byte is followed by the PHR data byte, indicating the frame length, followed by the PSDU data.

In Frame Buffer Access Mode during buffer reads, the PHY header (PHR) and the PSDU data are transferred via MISO following PHY_STATUS byte. Once the PSDU data is uploaded, three more bytes are transferred containing the link quality indication (LQI) value, the energy detection (ED) value, and the status information (RX_STATUS) of the received frame. The figure below illustrates the packet structure of a Frame Buffer read access.

Note: The frame buffer read access can be terminated immediately at any time by setting pin 23 (/SEL) = H, for example after reading the PHR byte only.

Figure 14-7. Packet Structure - Frame Read Access



The structure of RX_STATUS is described in the table below.

Table 14-3. Structure of RX_STATUS

Bit	7	6	5	4	
	RX_CRC_VALID	TRAC_STATUS			RX_STATUS
Read/Write	R	R	R	R	
Reset value	0	0	0	0	

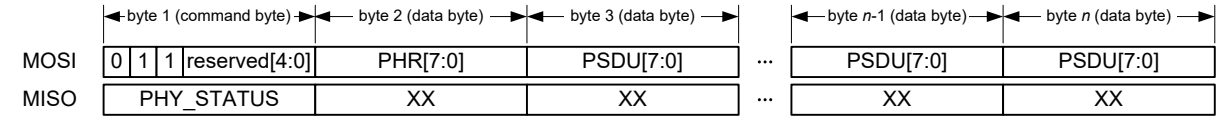
Bit	3	2	1	0	
	Reserved				RX_STATUS

Read/Write	R	R	R	R	
Reset value	0	0	0	0	

Note: More information on RX_CRC_VALID, see the PHY_RSSI register, and on TRAC_STATUS, see the TRX_STATE register.

On frame buffer write access, the second byte transferred on MOSI contains the frame length (PHR field) followed by the payload data (PSDU).

Figure 14-8. Packet Structure - Frame Write Access



The number of bytes *n* for one frame buffer access is calculated as follows:

Read Access *n* = 5 + *frame_length*

[PHY_STATUS, PHR byte, PSDU data, LQI, ED, and RX_STATUS]

Write Access : *n* = 2 + *frame_length*

[command byte, PHR byte, and PSDU data]

The maximum value of *frame_length* is 127 bytes. That means that *n* ≤ 132 for Frame Buffer read and *n* ≤ 129 for Frame Buffer write accesses.

Each read or write of a data byte automatically increments the address counter of the Frame Buffer until the access is terminated by setting /SEL = H. A Frame Buffer read access can be terminated at any time without any consequences by setting /SEL = H, for example after reading the frame length byte only. A successive Frame Buffer read operation starts again with the PHR field.

The content of the AT86RF212B Frame Buffer is overwritten by a new received frame or a Frame Buffer write access.

The figures below illustrate an exemplary SPI sequence of a Frame Buffer access to read a frame with 2-byte PSDU and write a frame with 4-byte PSDU.

Figure 14-9. Exemplary SPI Sequence - Frame Buffer Read of a Frame with 2-byte PSDU

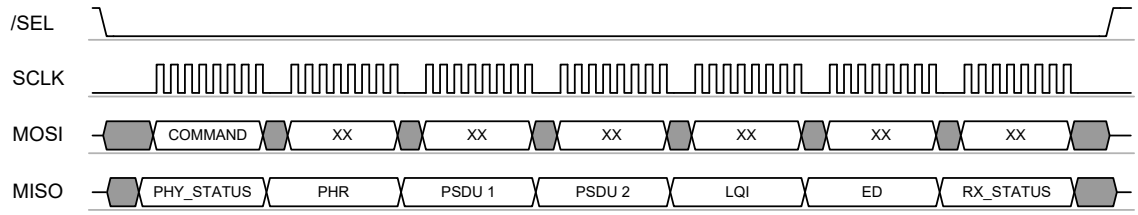
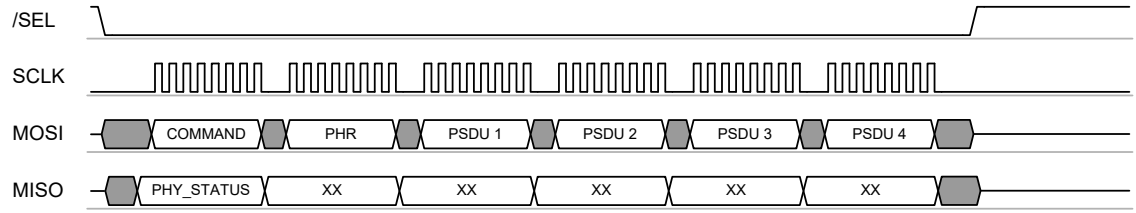


Figure 14-10. Exemplary SPI Sequence - Frame Buffer Write of a Frame with 4-byte PSDU



Access violations during a Frame Buffer read or write access are indicated by interrupt IRQ_6 (TRX_UR).

Notes:

1. The Frame Buffer is shared between RX and TX operations, the frame data is overwritten by freshly received data frames. If an existing TX payload data frame is to be retransmitted, it must be ensured that no TX data is overwritten by newly received RX data.
2. To avoid overwriting during receive *Dynamic Frame Buffer Protection* can be enabled.
3. For exceptions, receiving acknowledgement frames in Extended Operating Mode (TX_ARET).

Related Links

14.2.2.4 TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry
14.3.1 Introduction – IEEE 802.15.4-2006 Frame Format
14.3.8 Link Quality Indication (LQI)
14.4.4 Frame Buffer
14.6.7 Dynamic Frame Buffer Protection
14.8.6 PHY_RSSI
14.8.2 TRX_STATE

14.1.3.3 SRAM Access Mode

The SRAM access mode is used to read and write AT86RF212B frame buffer beginning with a specified byte address. It enables to access dedicated buffer data directly from a desired address without a need of incrementing the frame buffer from the top.

The SRAM access mode allows accessing dedicated bytes within the Frame Buffer or AES address space. This may reduce the SPI traffic.

During frame receive, after occurrence of IRQ_2 (RX_START), an SRAM access can be used to upload the PHR field while preserving Dynamic Frame Buffer Protection.

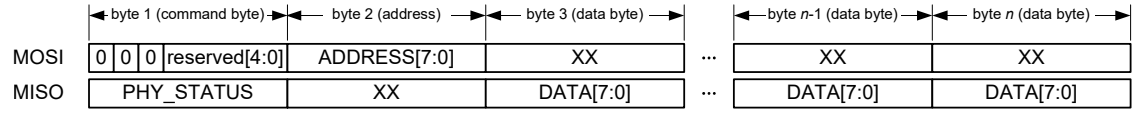
Each SRAM access starts with /SEL = L. The first transferred byte on MOSI shall be the command byte and must indicate an SRAM access mode according to the definition in SPI Command Byte Definition table. The following byte indicates the start address of the write or read access.

SRAM address space:

Frame Buffer	0x00 to 0x7F
AES	0x82 to 0x94

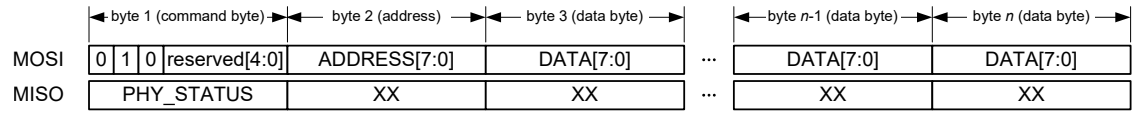
On SRAM read access, one or more bytes of read data are transferred on MISO starting with the third byte of the access sequence.

Figure 14-11. Packet Structure – SRAM Read Access



On SRAM write access, one or more bytes of write data are transferred on MOSI starting with the third byte of the access sequence. Do not attempt to read or write bytes beyond the SRAM buffer size.

Figure 14-12. Packet Structure – SRAM Write Access



As long as /SEL = L, every subsequent byte read or byte write increments the address counter of the Frame Buffer until the SRAM access is terminated by /SEL = H.

The figures below illustrate an exemplary SPI sequence of an AT86RF212B SRAM access to read and write a data package of five byte length, respectively.

Figure 14-13. Exemplary SPI Sequence – SRAM Read Access of a 5-byte Data Package

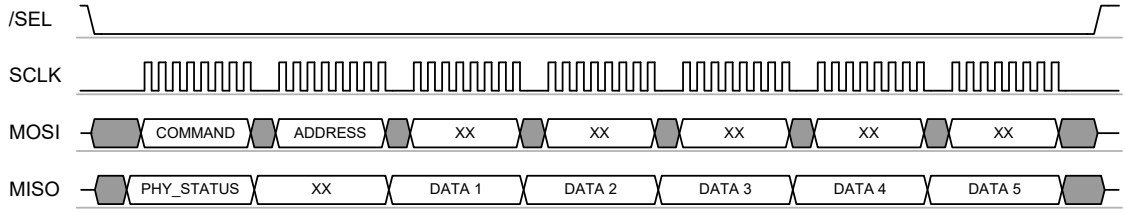
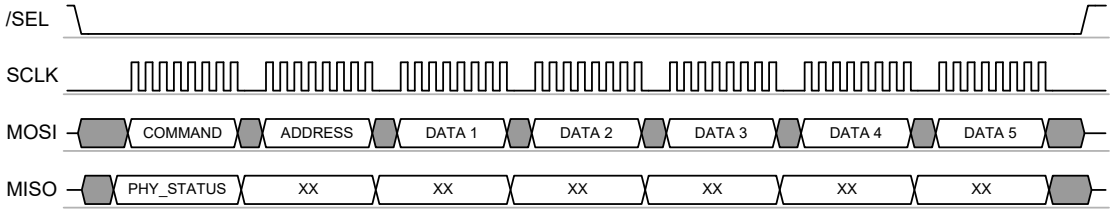


Figure 14-14. Exemplary SPI Sequence – SRAM Write Access of a 5-byte Data Package



Notes:

1. The SRAM access mode is not intended to be used as an alternative to the Frame Buffer access modes.
2. Frame Buffer access violations are not indicated by a TRX_UR interrupt when using the SRAM access mode.

Related Links

14.1.3 SPI Protocol
14.1.3.2 Frame Buffer Access Mode
14.4.8.5 Interrupt Handling
14.6.1 Security Module (AES)
14.6.7 Dynamic Frame Buffer Protection

14.1.4 Radio Transceiver Status Information

Each AT86RF212B SPI access returns the radio transceiver status information as the first byte transmitted by the MISO output while the serial data is shifted into the MOSI input. The radio transceiver status information (PHY_STATUS) can be configured using the SPI_CMD_MODE bits within the TRX_CTRL_1 register (TRX_CTRL_1.SPI_CMD_MODE) to return ether the TRX_STATUS, PHY_RSSI or IRQ_STATUS register.

Related Links

14.8.4 TRX_CTRL_1

14.1.5 Radio Transceiver Identification

The AT86RF212B can be identified by four registers. One 8-bit register contains a unique part number (PART_NUM) and one register contains the corresponding 8-bit version number (VERSION_NUM). Two additional 8-bit registers contain the JEDEC manufacture ID.

Related Links

14.8.27 PART_NUM
14.8.28 VERSION_NUM
14.8.29 MAN_ID_0
14.8.30 MAN_ID_1

14.1.6 Sleep/Wake-up and Transmit Signal (SLP_TR)

SLP_TR is a multi-functional pin. Its function relates to the current state of the AT86RF212B and is summarized in table below. The radio transceiver states are explained in detail in the *Operating Modes*.

Table 14-4. SLP_TR Multi-functional Pin

Transceiver Status	Function	Transition	Description
PLL_ON	TX start	L ⇨ H	Starts frame transmission

.....continued			
Transceiver Status	**Function**	**Transition**	**Description**
TX_ARET_ON	TX start	L ⇨ H	Starts TX_ARET transaction
BUSY_RX_AACK	TX start	L ⇨ H	Starts ACK transmission during RX_AACK slotted operation
TRX_OFF	Sleep	L ⇨ H	Takes the radio transceiver into SLEEP state; CLKM disabled
SLEEP	Wakeup	H ⇨ L	Takes the radio transceiver back into TRX_OFF state, level sensitive
RX_ON	Disable CLKM	L ⇨ H	Takes the radio transceiver into RX_ON_NOCLK state and disables CLKM
RX_ON_NOCLK	Enable CLKM	H ⇨ L	Takes the radio transceiver into RX_ON state and enables CLKM
RX_AACK_ON	Disable CLKM	L ⇨ H	Takes the radio transceiver into RX_AACK_ON_NOCLK state and disables CLKM
RX_AACK_ON_NOCLK	Enable CLKM	H ⇨ L	Takes the radio transceiver into RX_AACK_ON state and enables CLKM

In states PLL_ON and TX_ARET_ON, the signal SLP_TR is used as trigger input to initiate a TX transaction. Here SLP_TR is sensitive on rising edge only.

After initiating a state change by a rising edge at SLP_TR in radio transceiver states TRX_OFF, RX_ON or RX_AACK_ON, the radio transceiver remains in the new state as long as the pin is logical high and returns to the preceding state with the falling edge.

Related Links

14.2 Operating Modes

14.2.2.3.5 RX_AACK Slotted Operation – Slotted Acknowledgement

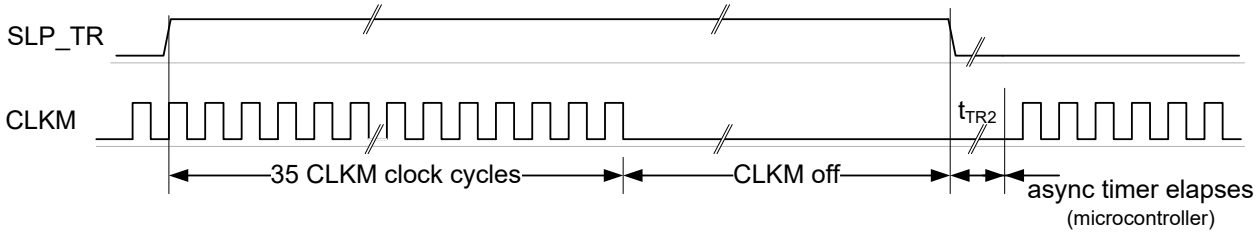
14.1.6.1 SLEEP State

The SLEEP state is used when radio transceiver functionality is not required, and thus the AT86RF212B can be powered down to reduce the overall power consumption.

A power-down scenario is shown in the figure below. When the radio transceiver is in TRX_OFF state, the microcontroller forces the AT86RF212B to SLEEP by setting SLP_TR = H. If pin 17 (CLKM) provides a clock to the microcontroller, this clock is switched off after 35 CLKM cycles. This enables a microcontroller in a synchronous system to complete its power-down routine and prevent deadlock situations. The AT86RF212B awakes when the microcontroller releases pin 11 (SLP_TR). This concept provides the lowest possible power consumption.

The CLKM clock frequency settings for CLKM_CTRL values six and seven are not intended to directly clock the microcontroller. When using these clock rates, CLKM is turned off immediately when entering SLEEP state.

Figure 14-15. Sleep and Wake-up Initiated by Asynchronous Microcontroller Timer



Note: Timing figures t_{TR3} and t_{TR1a} refer to the *State Transition Timing Summary*.

Related Links

14.2.1.4.6 State Transition Timing Summary

14.1.6.2 RX_ON and RX_AACK_ON States

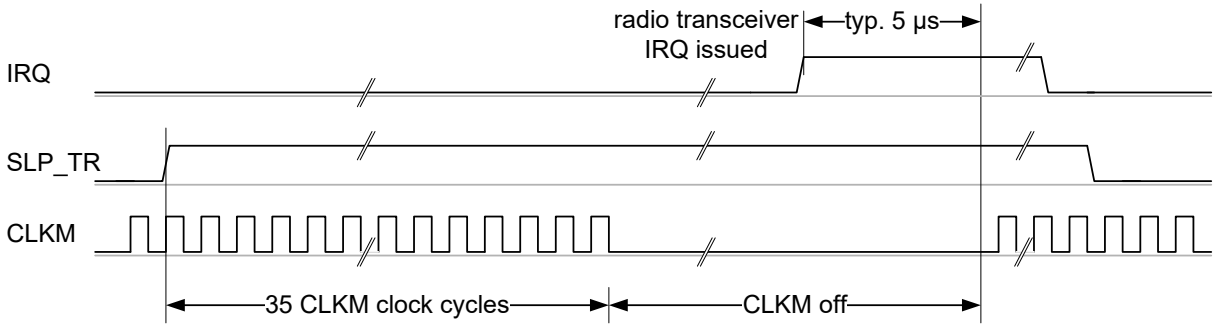
For synchronous systems where CLKM is used as a microcontroller clock source and the SPI master clock (SCLK) is directly derived from CLKM, the AT86RF212B supports an additional power-down mode for receive operating states (RX_ON and RX_AACK_ON).

If an incoming frame is expected and no other applications are running on the microcontroller, it can be powered down without missing incoming frames. This can be achieved by a rising edge on pin 11 (SLP_TR) that turns CLKM off. Then the radio transceiver state changes from RX_ON or RX_AACK_ON (Extended Operating Mode) to RX_ON_NOCLK or RX_AACK_ON_NOCLK, respectively. In case that a frame is received (for example indicated by an IRQ_2 (RX_START) interrupt), the clock output CLKM is automatically switched on again. This scenario is shown in the figure below. In RX_ON state, the clock at pin 17 (CLKM) is switched off after 35 CLKM cycles when setting SLP_TR = H.

The CLKM clock frequency settings for CLKM_CTRL values six and seven are not intended to directly clock the microcontroller. When using these clock rates, CLKM is turned off immediately when entering RX_ON_NOCLK or RX_AACK_ON_NOCLK.

In states RX_(AACK)_ON_NOCLK and RX_(AACK)_ON, the radio transceiver current consumptions are equivalent. However, the RX_(AACK)_ON_NOCLK current consumption is reduced by the current required for driving pin 17 (CLKM).

Figure 14-16. Wake-Up Initiated by Radio Transceiver Interrupt



14.1.7 Interrupt Logic

14.1.7.1 Overview

The AT86RF212B differentiates between nine interrupt events (eight physical interrupt registers, one shared by two functions). Each interrupt is enabled by setting the corresponding bit in the interrupt mask register 0x0E (IRQ_MASK). Internally, each pending interrupt is flagged in the interrupt status register. All interrupt events are OR-combined to a single external interrupt signal (IRQ pin). If an interrupt is issued pin 24 (IRQ) = H, the microcontroller shall read the interrupt status register 0x0F (IRQ_STATUS) to determine the source of the interrupt. A read access to this register clears the interrupt status register and thus the IRQ pin, too.

Interrupts are not cleared automatically when the event trigger for respective interrupt flag bit in the register 0x0F (IRQ_STATUS) is no longer active. Only a read access to register 0x0F (IRQ_STATUS) clears the flag bits. Exceptions are IRQ_0 (PLL_LOCK) and IRQ_1 (PLL_UNLOCK) where each is cleared in addition by the appearance of the other.

The supported interrupts for the Basic Operating Mode are summarized in table below.

Table 14-5. Interrupt Description in Basic Operating Mode

IRQ Name	Description
IRQ_7 (BAT_LOW)	Indicates a supply voltage below the programmed threshold.
IRQ_6 (TRX_UR)	Indicates a Frame Buffer access violation.
IRQ_5 (AMI)	Indicates address matching.

.....continued	
IRQ Name	**Description**
IRQ_4 (CCA_ED_DONE)	Multi-functional interrupt: 1. AWAKE_END: • Indicates radio transceiver reached TRX_OFF state at the end of P_ON ⇨ TRX_OFF and SLEEP ⇨ TRX_OFF state transition. 2. CCA_ED_DONE: • Indicates the end of a CCA or ED measurement.
IRQ_3 (TRX_END)	RX: Indicates the completion of a frame reception. TX: Indicates the completion of a frame transmission.
IRQ_2 (RX_START)	Indicates the start of a PSDU reception; the AT86RF212B state changed to BUSY_RX; the PHR can be read from Frame Buffer.
IRQ_1 (PLL_UNLOCK)	Indicates PLL unlock. If the radio transceiver is in BUSY_TX / BUSY_TX_ARET state, the PA is turned off immediately.
IRQ_0 (PLL_LOCK)	Indicates PLL lock.

Note: The IRQ_4 (AWAKE_END) interrupt can usually not be seen when the transceiver enters TRX_OFF state after P_ON or RESET, because register 0x0E (IRQ_MASK) is reset to mask all interrupts. It is recommended to enable IRQ_4 (AWAKE_END) to be notified once the TRX_OFF state is entered.

Related Links

14.2.1.3 Interrupt Handling
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.3.2 Frame Filter
14.3.6.4 Interrupt Handling
14.4.4.3 Interrupt Handling
14.4.6.4 Interrupt Handling
14.4.8.5 Interrupt Handling
14.2.2.5 Interrupt Handling
14.8.15 IRQ_STATUS

14.1.7.2 Interrupt Mask Modes and Pin Polarity

If the IRQ_MASK_MODE bits in the TRX_CTRL_1 register (TRX_CTRL_1.IRQ_MASK_MODE) is set, an interrupt event can be read from IRQ_STATUS register even if the interrupt itself is masked. However, in that case no timing information for this interrupt is provided.

Table 14-6. IRQ Mask Configuration

IRQ_MASK Value	IRQ_MASK_MODE	Description
0	0	IRQ is suppressed entirely and none of interrupt sources are shown in register IRQ_STATUS.
0	1	IRQ is suppressed entirely but all interrupt causes are shown in register IRQ_STATUS.
≠ 0	0	All enabled interrupts are signaled on IRQ pin and are also shown in register IRQ_STATUS.
≠ 0	1	All enabled interrupts are signaled on IRQ pin and all interrupt causes are shown in register IRQ_STATUS.

Figure 14-17. IRQ_MASK_MODE = 0

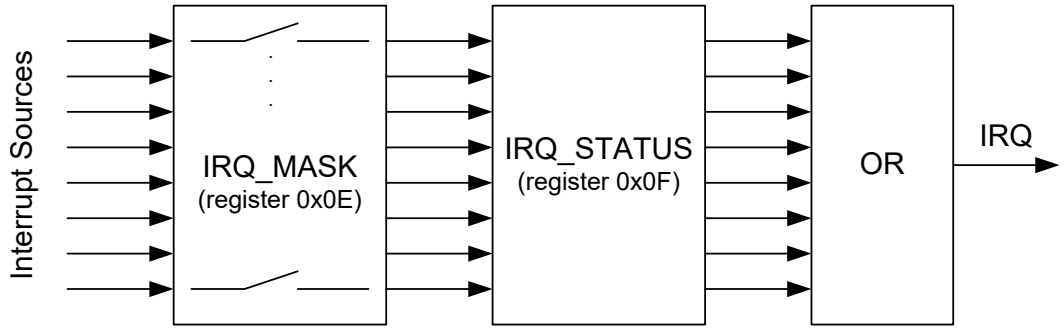
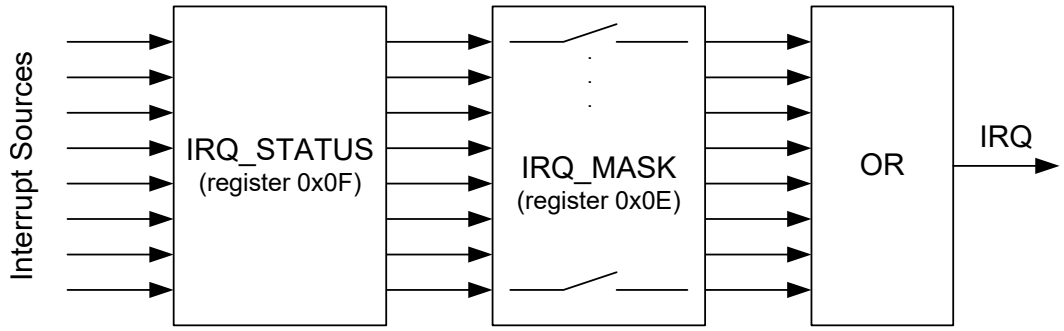


Figure 14-18. IRQ_MASK_MODE = 1



The AT86RF212B IRQ pin polarity can be configured with the IRQ_POLARITY bit in the TRX_CTRL_1 register (TRX_CTRL_1.IRQ_POLARITY). The default behavior is active high, which means that pin 24 (IRQ) = H issues an interrupt request.

If the “Frame Buffer Empty Indicator” is enabled during Frame Buffer read access, the IRQ pin has an alternative functionality.

A solution to monitor the IRQ_STATUS register (without clearing it) is described in Section BBD.

Related Links

14.6.6 Frame Buffer Empty Indicator

14.8.15 IRQ_STATUS

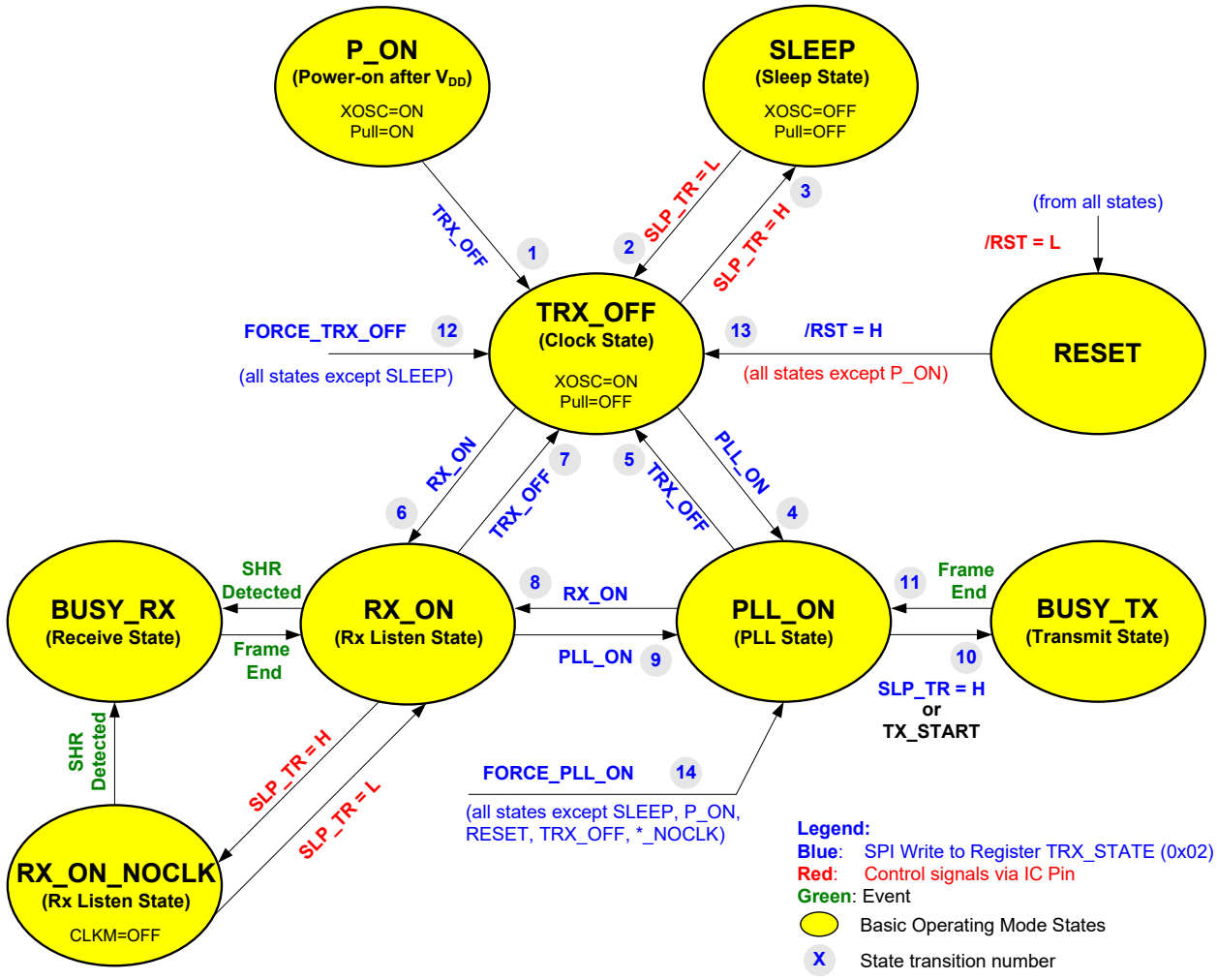
14.8.4 TRX_CTRL_1

14.2 Operating Modes

14.2.1 Operating Modes

This section summarizes all states to provide the basic functionality of AT86RF212B, such as receiving and transmitting frames, the power-on sequence, and sleep. The Basic Operating Mode is designed for IEEE 802.15.4 and general ISM band applications; the corresponding radio transceiver states are shown in the figure below.

Figure 14-19. Basic Operating Mode State Diagram



14.2.1.1 State Control

The radio transceiver's states are controlled by shifting serial digital data using the SPI to write individual commands to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). Change of the transceiver state can also be triggered by driving directly two signal pins: pin 11 (SLP_TR) and pin 8 (/RST). A successful state change can be verified by reading the radio transceiver status from register bits TRX_STATUS (register 0x01, TRX_STATUS).

If TRX_STATUS = 0x1F (STATE_TRANSITION_IN_PROGRESS), the AT86RF212B is in a state transition. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS.

Pin 11 (SLP_TR) is a multifunctional pin. Dependingon the radio transceiver state, a rising edge of pin 11 (SLP_TR) causes the following state transitions:

- TRX_OFF ⇨ SLEEP (level sensitive)
- RX_ON ⇨ RX_ON_NOCLK (level sensitive)
- PLL_ON ⇨ BUSY_TX

Whereas the falling edge of pin SLP_TR causes the following state transitions:

- SLEEP ⇨ TRX_OFF (level sensitive)
- RX_ON_NOCLK ⇨ RX_ON

A low level on pin 8 (/RST) causes a reset of all registers (register bits CLKM_CTRL are shadowed) and forces the radio transceiver into TRX_OFF state. However, if the device was in P_ON state it remains in the P_ON state.

For all states except SLEEP, the state change commands FORCE_TRX_OFF or TRX_OFF lead to a transition into TRX_OFF state. If the radio transceiver is in active receive or transmit states (BUSY_*), the command FORCE_TRX_OFF interrupts these active processes, and forces an immediate transition to TRX_OFF. In contrast a TRX_OFF command is stored until an active state (receiving or transmitting) has been finished. After that the transition to TRX_OFF is performed.

For a fast transition from any non sleep states to PLL_ON state the command FORCE_PLL_ON is provided. Active processes are interrupted. In contrast to FORCE_TRX_OFF, this command does not disable the PLL and the analog voltage regulator (AVREG). It is not available in states P_ON, SLEEP, RESET, and all *_NOCLK states.

The completion of each requested state change shall always be confirmed by reading the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS).

Note: If FORCE_TRX_OFF and FORCE_PLL_ON commands are used, it is recommended to set pin 11 (SLP_TR) = L before.

Related Links

14.1.6 Sleep/Wake-up and Transmit Signal (SLP_TR)
14.8.1 TRX_STATUS

14.2.1.2 Basic Operating Mode Description

14.2.1.2.1 P_ON – Power-On after V_{DD}

When the external supply voltage (V_{DD}) is applied first to the AT86RF212B, the radio transceiver goes into P_ON state performing an on-chip reset. The crystal oscillator is activated and the default 1MHz master clock is provided at pin 17 (CLKM) after the crystal oscillator has stabilized. CLKM can be used as a clock source to the microcontroller. The SPI interface and digital voltage regulator (DVREG) are enabled.

The on-chip power-on-reset sets all registers to their default values. A dedicated reset signal from the microcontroller at pin 8 (/RST) is not necessary, but recommended for hardware / software synchronization reasons.

All digital inputs are pulled-up or pulled-down during P_ON state. This is necessary to support microcontrollers where GPIO signals are floating after power-on or reset. The input pull-up and pull-down transistors are disabled when the radio transceiver leaves P_ON state towards TRX_OFF state. A reset during P_ON state does not change the pull-up and pull-down configuration.

Leaving P_ON state, output pins DIG1/DIG2 are pulled-down to digital ground, whereas pins DIG3/DIG4 are pulled-down to analog ground, unless their configuration is changed.

Prior to leaving P_ON, the microcontroller must set the AT86RF212B pins to the default operating values: pin 11 (SLP_TR) = L, pin 8 (/RST) = H and pin 23 (/SEL) = H.

All interrupts are disabled by default. Thus, interrupts for state transition control are to be enabled first, for example enable IRQ_4 (AWAKE_END) to indicate a state transition to TRX_OFF state or interrupt IRQ_0 (PLL_LOCK) to signal a locked PLL in PLL_ON state. In P_ON state a first access to the radio transceiver registers is possible after a default 1MHz master clock is provided at pin 17 (CLKM), refer to t_{TR1} in the *State Transition Timing Summary*.

Once the supply voltage has stabilized and the crystal oscillator has settled (see parameter t_{XTAL} refer to Table 7-2), the interrupt mask for the AWAKE_END should be set. A valid SPI write access to TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRC_CMD) with the command TRX_OFF or FORCE_TRX_OFF initiate a state change from P_ON towards TRX_OFF state, which is then indicated by an interrupt IRQ_4 (AWAKE_END) if enabled.

14.2.1.2.2 SLEEP - Sleep State

In SLEEP state, the radio transceiver is disabled. The radio transceiver current consumption is reduced to leakage current plus the current of a low power voltage regulator (typ. 100nA). This regulator provides the supply voltage to the registers to preserve their contents. SLEEP state can only be entered from state TRX_OFF, by setting SLP_TR = H.

If CLKM is enabled with a clock rates higher than 250kHz, the SLEEP state is entered 35 CLKM cycles after the rising edge at pin 11 (SLP_TR). At that time CLKM is turned off. If the CLKM output is already turned off (register bits CLKM_CTRL = 0), the SLEEP state is entered immediately.

At clock rates of 250kHz and symbol clock rate (TRX_CTRL_0.CLKM_CTRL values six and seven), the main clock at pin 17 (CLKM) is turned off immediately.

Setting SLP_TR = L returns the radio transceiver back to the TRX_OFF state. During SLEEP state the radio transceiver register contents and the AES register contents remain valid while the contents of the Frame Buffer are lost.

/RST = L in SLEEP state returns the radio transceiver to TRX_OFF state and thereby sets all registers to their default values. Exceptions are the CLKM_CTRL bits in the TRX_CTRL_0 register (TRX_CTRL_0.CLKM_CTRL). These register bits require a specific treatment.

Related Links

14.8.3 TRX_CTRL_0

14.2.1.2.3 TRX_OFF - Clock State

In TRX_OFF the crystal oscillator is running and the master clock is available if enabled. The SPI interface and digital voltage regulator are enabled, thus the radio transceiver registers, the Frame Buffer and security engine (AES) are accessible.

In contrast to P_ON state the pull-up and pull-down configuration is disabled.

Notes:

1. Pin 11 (SLP_TR) and pin 8 (/RST) are available for state control.
2. The analog front-end is disabled during TRX_OFF state. If the TRX_OFF_AVDD_EN bit in the TRX_CTRL_2 register (TRX_CTRL_2.TRX_OFF_AVDD_EN) is set, the analog voltage regulator is turned on, enabling faster switch to any transmit/receive state.

Entering the TRX_OFF state from P_ON, SLEEP, or RESET state, the state change is indicated by interrupt IRQ_4 (AWAKE_END) if enabled.

Related Links

14.8.12 TRX_CTRL_2

14.2.1.2.4 PLL_ON - PLL State

Entering the PLL_ON state from TRX_OFF state enables the analog voltage regulator (AVREG) first, unless the AVREG is already switched on (TRX_CTRL_2.TRX_OFF_AVDD_EN). After the voltage regulator has been settled, the PLL frequency synthesizer is enabled. When the PLL has been settled at the receive frequency to a channel defined by the CHANNEL bits in the PHY_CC_CCA register (PHY_CC_CCA.CHANNEL) or the CC_NUMBER bits in the CC_CTRL_0 register (CC_CTRL_0.CC_NUMBER) and the CC_BAND bits in the CC_CTRL_1 register (CC_CTRL_1.CC_BAND), a successful PLL lock is indicated by issuing an interrupt IRQ_0 (PLL_LOCK).

If an RX_ON command is issued in PLL_ON state, the receiver is enabled immediately. If the PLL has not been settled before the state change nevertheless takes place. Even if the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS) indicates RX_ON, actual frame reception can only start once the PLL has locked.

The PLL_ON state corresponds to the TX_ON state in IEEE 802.15.4.

Related Links

14.8.19 CC_CTRL_0
14.8.20 CC_CTRL_1
14.8.8 PHY_CC_CCA
14.8.1 TRX_STATUS

14.2.1.2.5 RX_ON and BUSY_RX - RX Listen and Receive State

In RX_ON state the receiver is in the RX data polling mode and the PLL frequency synthesizer is locked to its preprogrammed frequency.

The AT86RF212B receive mode is internally separated into RX_ON state and BUSY_RX state. There is no difference between these states with respect to the analog radio transceiver circuitry, which are always turned on. In both states, the receiver and the PLL frequency synthesizer are enabled.

During RX_ON state, the receiver listens for incoming frames. After detecting a valid synchronization header (SHR), the AT86RF212B automatically enters the BUSY_RX state. The reception of a valid PHY header (PHR) generates an IRQ_2 (RX_START) if enabled.

During PSDU reception, the frame data are stored continuously in the Frame Buffer until the last byte was received. The completion of the frame reception is indicated by an interrupt IRQ_3 (TRX_END) and the radio transceiver reenters the state RX_ON. At the same time the RX_CRC_VALID bits in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID) is updated with the result of the FCS check.

Received frames are passed to the frame filtering unit. If the content of the MAC addressing fields (refer to [2] IEEE 802.15.4-2006, Section 7.2.1) generates a match, IRQ_5 (AMI) interrupt is issued. The expected address values are to be stored in registers 0x20 – 0x2B (Short address, PAN-ID and IEEE address). Frame filtering is available in Basic Operating Mode and Extended Operating Mode.

Leaving state RX_ON is possible by writing a state change command to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD).

Related Links

14.8.6 PHY_RSSI
14.8.2 TRX_STATE
14.1.7 Interrupt Logic

14.2.1.2.6 RX_ON_NOCLK - RX Listen State without CLKM

In RX_ON_NOCLK state the receiver is in the RX data polling mode with CLKM output disabled.

If the radio transceiver is listening for an incoming frame and the microcontroller is not running an application, the microcontroller may be powered down to decrease the total system power consumption. This specific power-down scenario – for systems running in clock synchronous mode – is supported by the AT86RF212B using the state RX_ON_NOCLK.

This state can only be entered by asserting pin 11 (SLP_TR) = H while the radio transceiver is in RX_ON state. Pin 17 (CLKM) is disabled 35 CLKM cycles after the rising edge at pin 11 (SLP_TR). This allows the microcontroller to complete its power-down sequence(1).

Once in RX_ON_NOCLK state a valid SHR header triggers a state transition to BUSY_RX state. The reception of a frame shall be indicated to the microcontroller by an interrupt indicating the receive status. CLKM is turned on again, and the radio transceiver enters the BUSY_RX state. When using RX_ON_NOCLK, it is essential to enable at least one interrupt request indicating the reception status.

After the receive transaction has been completed, the radio transceiver enters the RX_ON state. The radio transceiver only reenters the RX_ON_NOCLK state when the next rising edge of pin 11 (SLP_TR) occurs.

If the AT86RF212B is in the RX_ON_NOCLK state and pin 11 (SLP_TR) is reset to logic low, it enters the RX_ON state and it starts to supply clock on pin 17 (CLKM) again(2).

Notes:

1. For CLKM clock rates 250kHz and symbol clock rates (TRX_CTRL_0.CLKM_CTRL values six and seven), the master clock signal CLKM is switched off immediately after the rising edge of pin 11 (SLP_TR).
2. A reset in state RX_ON_NOCLK further requires to reset pin 11 (SLP_TR) to logic low, otherwise the radio transceiver enters directly the SLEEP state.

Related Links

14.1 Microcontroller Interface
14.1.6 Sleep/Wake-up and Transmit Signal (SLP_TR)
14.8.3 TRX_CTRL_0

14.2.1.2.7 BUSY_TX - Transmit State

In the BUSY_TX state AT86RF212B is in the data transmission state.

A transmission can only be initiated from the PLL_ON state. The transmission can be started either by driving event such as:

- A rising edge on pin 11 (SLP_TR), or
- A serial TX_START command via the SPI to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD).

Either of these forces the radio transceiver into the BUSY_TX state.

During the transition to the BUSY_TX state, the PLL frequency shifts to the transmit frequency. The actual transmission of the first data chip of the SHR starts after one symbol period (see note) in order to allow PLL settling and PA ramp-up. After transmission of the SHR, the Frame Buffer content is transmitted. In case the PHR indicates a frame length of zero, the transmission is aborted immediately after the PHR field.

After the frame transmission has been completed, the AT86RF212B automatically turns off the power amplifier, generates an IRQ_3 (TRX_END) interrupt, and returns into PLL_ON state.

Note: Throughout this datasheet, a “symbol period” refers to the definition described in the *Symbol Period* section.

Related Links

14.8.2 TRX_STATE

14.2.1.2.8 RESET State

The RESET state is used to set back the state machine and to reset all registers of AT86RF212B to their default values; exceptions are register bits CLKM_CTRL bits in the TRX_CTRL_0 register (TRX_CTRL_0.CLKM_CTRL). These register bits require a specific treatment, for details see section *Master Clock Signal Output (CLKM)*.

Once in RESET state a device enters TRX_OFF state by setting pulling a reset pin high pin 8 (/RST) = H. If the device is still in the P_ON state it remains in the P_ON state though. A reset is triggered by pulling /RST pin low pin 8 (/RST) = L and the state returns after setting /RST = H. The reset pulse should have a minimum length as specified in the sections *Reset procedures* and *Digital Interface Timing Characteristics* (parameter t_{10}). During reset, the microcontroller has to set the radio transceiver control pins SLP_TR and /SEL to their default values.

Related Links

14.8.3 TRX_CTRL_0

14.2.1.3 Interrupt Handling

All interrupts provided by the AT86RF212B are supported in Basic Operating Mode. For example, interrupts are provided to observe the status of radio transceiver RX and TX operations.

When being in receive mode, IRQ_2 (RX_START) indicates the detection of a valid PHR first, IRQ_5 (AMI) an address match, and IRQ_3 (TRX_END) the completion of the frame reception. During transmission, IRQ_3 (TRX_END) indicates the completion of the frame transmission.

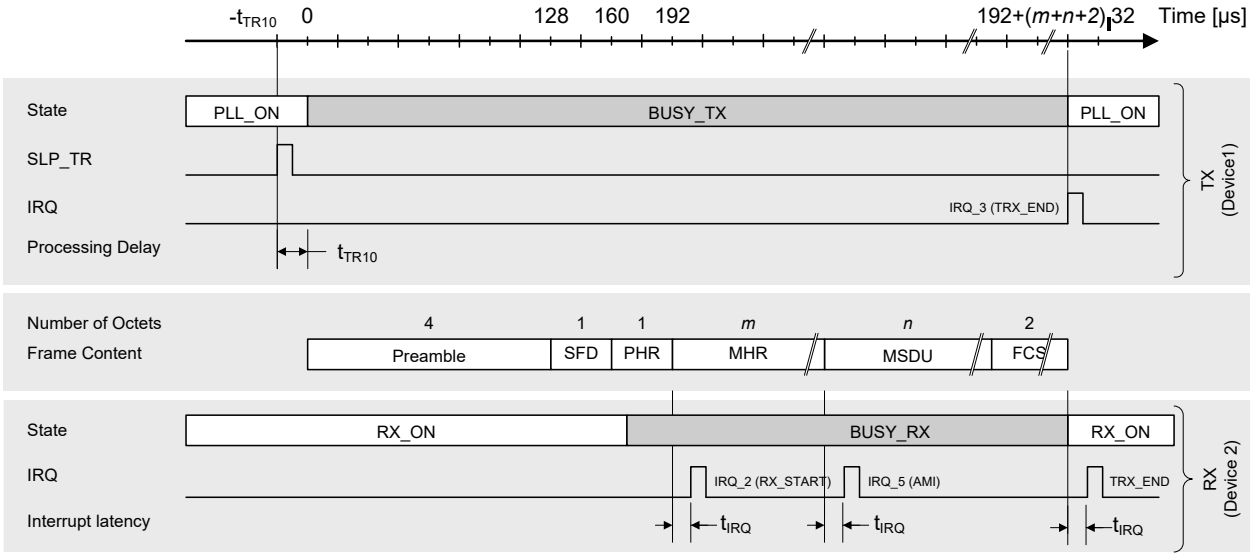
The figure below shows an example for a transmit/receive transaction between two devices and the related interrupt events in Basic Operating Mode. Device 1 transmits a frame containing a MAC header (in this example of length seven), MAC payload, and a valid FCS. The end of the frame transmission is indicated by IRQ_3 (TRX_END).

The frame is received by Device 2. Interrupt IRQ_2 (RX_START) indicates the detection of a valid PHR field and IRQ_3 (TRX_END) the completion of the frame reception. If the frame passes the Frame Filter, an address match interrupt IRQ_5 (AMI) is issued after the reception of the MAC header (MHR). The received frame is stored in the Frame Buffer.

In Basic Operating Mode the third interrupt IRQ_3 (TRX_END) is issued at the end of the received frame. In Extended Operating Mode the interrupt is only issued if the received frame passes the address filter and the FCS is valid.

Processing delay t_{IRQ} is a typical value.

Figure 14-20. Timing of RX_START, AMI and TRX_END Interrupts in Basic Operating Mode for O-QPSK 250kb/s Mode



Related Links

14.1.7 Interrupt Logic

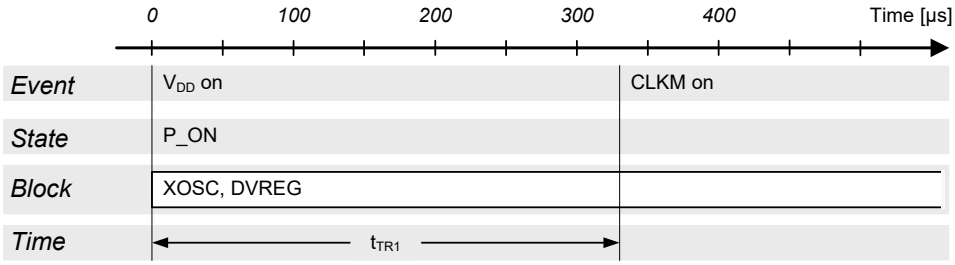
14.2.1.4 Basic Operating Mode Timing

This section depicts AT86RF212B state transitions and their timing properties.

14.2.1.4.1 Power-on Procedure

The power-on procedure to P_ON state is shown in the figure below.

Figure 14-21. Power-on Procedure to P_ON State

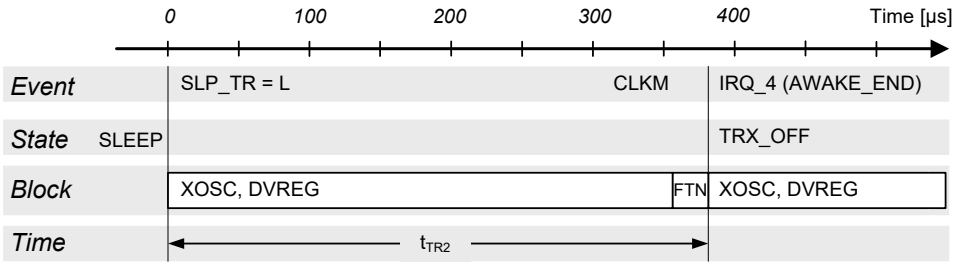


When the external supply voltage (V_{DD}) is initially supplied to the AT86RF212B, the radio transceiver enables the crystal oscillator (XOSC) and the internal 1.8V voltage regulator for the digital domain (DVREG). After t_{TR1} = 420µs (typ.), the master clock signal is available at pin 17 (CLKM) at default rate of 1MHz. As soon as CLKM is available the SPI is enabled and can be used to control the transceiver. As long as no state change towards state TRX_OFF is performed, the radio transceiver remains in P_ON state.

14.2.1.4.2 Wake-up Procedure from SLEEP

The wake-up procedure from SLEEP state is shown in the figure below.

Figure 14-22. Wake-up Procedure from SLEEP State



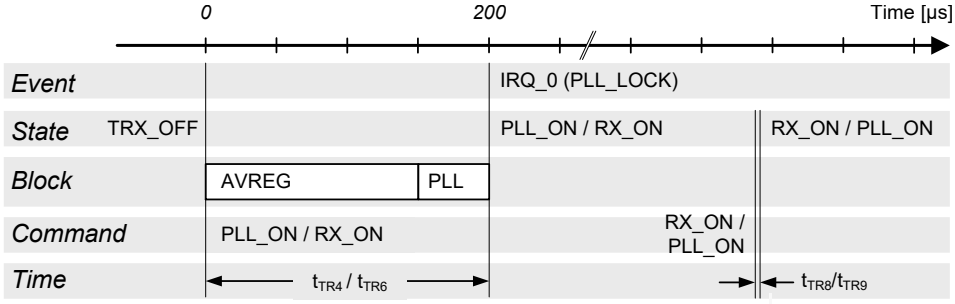
The radio transceiver's SLEEP state is left by releasing pin 11 (SLP_TR) to logic low. This restarts the XOSC and DVREG. After t_{TR2} = 420µs (typ.) the radio transceiver enters TRX_OFF state. The internal clock signal is available and provided to pin 17 (CLKM), if enabled.

This procedure is similar to the Power-on Procedure. However the radio transceiver automatically proceeds to the TRX_OFF state. During this, transition the filter-tuning network (FTN) calibration is performed. Entering TRX_OFF state is signaled by IRQ_4 (AWAKE_END), if this interrupt was enabled by the appropriate mask register bit.

14.2.1.4.3 PLL_ON and RX_ON States

The transition from TRX_OFF to PLL_ON or RX_ON state and further to RX_ON or PLL_ON is shown in the figure below.

Figure 14-23. Transition from TRX_OFF to PLL_ON/RX_ON State and further to RX_ON/PLL_ON



Notes:

1. If TRX_CMD = RX_ON in TRX_OFF state, RX_ON state is entered immediately, even if the PLL has not settled.
2. Timing figures t_{TR4}, t_{TR6}, t_{TR8}, and t_{TR9} refers to the *State Transition Timing Summary*.

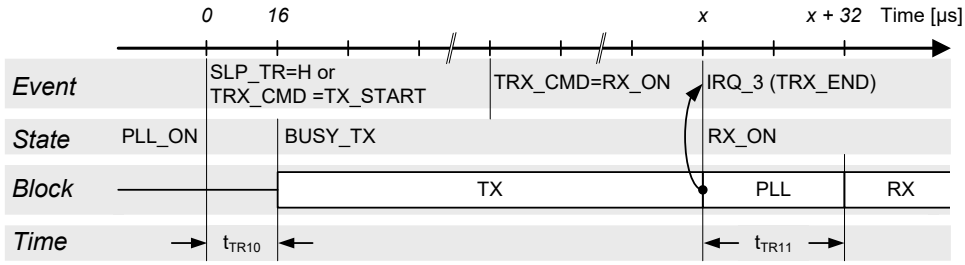
In TRX_OFF state, entering the commands PLL_ON or RX_ON initiates a ramp-up sequence of the internal 1.8V voltage regulator for the analog domain (AVREG). RX_ON state can be entered any time from PLL_ON state, regardless whether the PLL has already locked, which is indicated by IRQ_0 (PLL_LOCK). Likewise, PLL_ON state can be entered any time from RX_ON state.

When TRX_OFF_AVDD_EN (register 0x0C, TRX_CTRL_2) is already set in TRX_OFF state, the analog voltage regulator is turned on immediately and the ramp-up sequence to PLL_ON or RX_ON can be accelerated.

14.2.1.4.4 BUSY_TX to RX_ON States

The transition from PLL_ON to BUSY_TX state and subsequently to RX_ON state is shown in the figure below.

Figure 14-24. PLL_ON to BUSY_TX to RX_ON Timing for O-QPSK 250kb/s Mode



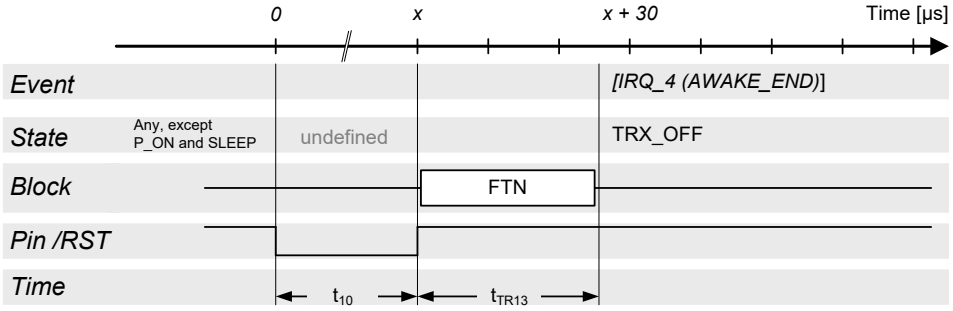
Starting from PLL_ON state, it is further assumed that the PLL has already been locked. A transmission is initiated either by a rising edge of pin 11 (SLP_TR) or by command TX_START. The PLL settles to the transmit frequency and the PA is enabled. After the duration of t_{TR10} (one symbol period), the AT86RF212B changes into BUSY_TX state, transmitting the internally generated SHR and the PSDU data of the Frame Buffer. After completing the frame transmission, indicated by IRQ_3 (TRX_END), the PLL settles back to the receive frequency within t_{TR11} = 32µs and returns to state PLL_ON.

If during BUSY_TX the radio transmitter is requested to change to a receive state, it automatically proceeds to state RX_ON upon completion of the transmission.

14.2.1.4.5 Reset Procedure

The radio transceiver reset procedure is shown in the figure below.

Figure 14-25. Reset Procedure



1. Timing figure t_{TR13} refers to *State Transition Timing* table, t_{10} refers to the *Digital Interface Timing Characteristics*.

/RST = L sets all registers to their default values. Exceptions are the CLKM_CTRL bits in the TRX_CTRL_0 register (TRX_CTRL_0.CLKM_CTRL). After releasing the reset pin 8 (/RST) = H, the wake-up sequence including an FTN calibration cycle is performed. After that, the TRX_OFF state is entered.

The above figure illustrates the reset procedure once P_ON state was left and the radio transceiver was not in SLEEP state.

The reset procedure is identical for all originating radio transceiver states except of states P_ON and SLEEP. Instead, the procedures described in the *Basic Operation State Control* and *Basic Operationg Mode Description* sections must be followed to enter the TRX_OFF state.

If the radio transceiver was in SLEEP state, the XOSC and DVREG are enabled before entering TRX_OFF state.

If register bits TRX_STATUS indicates STATE_TRANSITION_IN_PROGRESS during system initialization until the AT86RF212B reaches TRX_OFF state, do not try to initiate a further state change while the radio transceiver is in this state.

Notes:

1. The reset impulse should have a minimum length t_{10} = 625ns as specified in the *Digital Interface Timing Characteristics*.
2. An access to the device should not occur earlier than $t_{11} \geq$ 625ns after releasing the /RST pin; refer the *Digital Interface Timing Characteristics*.
3. A reset overrides an SPI command request that might have been queued.

Related Links

14.2.1.2 Basic Operating Mode Description
14.2.1.1 State Control
14.2.1.4.6 State Transition Timing Summary
14.8.3 TRX_CTRL_0

14.2.1.4.6 State Transition Timing Summary

The AT86RF212B transition numbers correspond to the *Basic Operation Mode State Diagram* and do not include SPI access time if not otherwise stated. See measurement setup in the *Basic Application Schematic*.

Table 14-7. State Transition Timing

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{TR1}	P_ON⇒CLKM is available	Depends on crystal oscillator setup (Siward A207-011, CL= 10pF) and external capacitor at DVDD (CB3 = 1µF nom.).		420	1000	µs
t_{TR1a}	SLEEP⇒CLKM is available	Depends on crystal oscillator setup (Siward A207-011, CL= 10pF) and external capacitor at DVDD (CB3 = 1µF nom.).		390	1000	µs
t_{TR2}	SLEEP⇒TRX_OFF	Depends on crystal oscillator setup (Siward A207-011, CL= 10pF) and external capacitor at DVDD (CB3 = 1µF nom.); TRX_OFF state indicated by IRQ_4 (AWAKE_END).		420	1000	µs
t_{TR3}	TRX_OFF⇒SLEEP	For f_{CLKM} > 250kHz.		35		CLKM cycles
		Otherwise.		0		CLKM cycles
t_{TR4}	TRX_OFF⇒PLL_ON	Depends on external capacitor at AVDD (CB1 = 1µF nom.); The TRX_OFF_AVDD_EN bit in the TRX_CTRL_2 register (TRX_CTRL_2.TRX_OFF_AVDD_EN) is not set.		170		µs
t_{TR5}	PLL_ON⇒TRX_OFF			1		µs
t_{TR6}	TRX_OFF⇒RX_ON	Depends on external capacitor at AVDD (CB1 = 1µF nom.); The TRX_OFF_AVDD_EN bit in the TRX_CTRL_2 register (TRX_CTRL_2.TRX_OFF_AVDD_EN) is not set.		170		µs
t_{TR7}	RX_ON⇒TRX_OFF			1		µs
t_{TR8}	PLL_ON⇒RX_ON			1		µs
t_{TR9}	RX_ON⇒PLL_ON	Transition time is also valid for TX_ARET_ON, RX_AACK_ON⇒PLL_ON.		1		µs
t_{TR10}	PLL_ON⇒BUSY_TX	When asserting pin 11 (SLP_TR) or TRX_CMD = TX_START first symbol transmission is delayed by one symbol period (PLL settling and PA ramp-up).		1		symbol period
t_{TR11}	BUSY_TX⇒PLL_ON	PLL settling time.		32		µs

.....continued						
Symbol	**Parameter**	**Condition**	**Min.**	**Typ.**	**Max.**	**Unit**
t_{TR12}	Various states⇒TRX_OFF	Using TRX_CMD = FORCE_TRX_OFF (see TRX_STATE.TRX_CMD register bits); not valid for SLEEP⇒TRX_OFF (see t_{TR2}).		1		µs
t_{TR13}	RESET⇒TRX_OFF	Not valid for P_ON or SLEEP.		26		µs
t_{TR14}	Various states⇒PLL_ON	Using TRX_CMD = FORCE_PLL_ON (see TRX_STATE.TRX_CMD register bits); not valid for states SLEEP, P_ON, RESET, TRX_OFF, and *_NOCLK.		1		µs

The state transition timing is calculated based on the timing of the individual blocks shown in the *Basic Operating Mode Timing* section. The worst case values include maximum operating temperature, minimum supply voltage, and device parameter variations.

Table 14-8. Block Initialization and Settling Time

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{XTAL}	Reference oscillator settling time	Start XTALÞclock available at pin 17 (CLKM). Depends on crystal oscillator setup (Siward A207-011, CL= 10pF).		420	1000	µs
t_{FTN}	FTN calibration time				25	µs
t_{DVREG}	DVREG settling time	Depends on external bypass capacitor at DVDD (CB3 = 1µF nom., 10µF worst case).		150	1500	µs
t_{AVREG}	AVREG settling time	Depends on external bypass capacitor at AVDD (CB1 = 1µF nom., 10µF worst case).		150	1500	µs
t_{PLL_INIT}	Initial PLL settling time	PLL settling time TRX_OFF⇒PLL_ON, including 150µs AVREG settling time.		170	370	µs
t_{PLL_SW}	PLL settling time on channel switch	Duration of channel switch within frequency band.		11	42	µs
t_{PLL_CF}	PLL CF calibration	PLL center frequency calibration.	8	8	270	µs
t_{PLL_DCU}	PLL DCU calibration	PLL DCU calibration.		10	10	µs
t_{RX_TX}	RX⇒TX	Maximum settling time RX⇒TX.			16	µs
t_{TX_RX}	TX⇒RX	Maximum settling time TX⇒RX.			32	µs
t_{RSSI}	RSSI, update	RSSI update period in receive states.				
		BPSK-20:		32		µs
		BPSK-40:		24		µs
		O-QPSK:		8		µs
t_{ED}	ED measurement	ED measurement period is eight symbols. Different timing within High Data Rate Modes.		8		symbol
t_{CCA}	CCA measurement	CCA measurement period is eight symbols.		8		symbol
t_{RND}	Random value, update	Random value update period.		1		µs
t_{AES}	AES core cycle time			23.4	24	µs

Related Links

14.2.1 Operating Modes
14.2.1.4 Basic Operating Mode Timing
14.8.12 TRX_CTRL_2
14.8.2 TRX_STATE

14.2.2 Extended Operationg Mode

Extended Operating Mode makes up for a large set of automated functionality add-ons which can be referred to as a hardware MAC accelerator. These add-ons go beyond the basic radio transceiver functionality provided by the Basic Operating Mode. Extended Operating Mode functions handle time critical MAC tasks, requested by the IEEE 802.15.4 standard, in hardware, such as automatic acknowledgement, automatic CSMA-CA, and retransmission. This results in a more efficient IEEE 802.15.4 software MAC implementation, including reduced code size, and may allow use of a smaller microcontroller or operation at low clock rates.

The Extended Operating Mode is designed to support IEEE 802.15.4-2006 and IEEE 802.15.4-2011 compliant frames; the mode is backward compatible to IEEE 802.15.4-2003 and supports non IEEE 802.15.4 compliant frames. This mode comprises the following procedures:

Automatic acknowledgement (RX_AACK) divides into the tasks:

- Frame reception and automatic FCS check
- Configurable addressing fields check
- Interrupt indicating address match
- Interrupt indicating frame reception, if it passes address filtering and FCS check
- Automatic ACK frame transmission (if the received frame passed the address filter and FCS check and if an ACK is required by the frame type and ACK request)
- Support of slotted acknowledgment using SLP_TR pin (used for beacon-enabled operation)

Automatic CSMA-CA and Retransmission (TX_ARET) divides into the tasks:

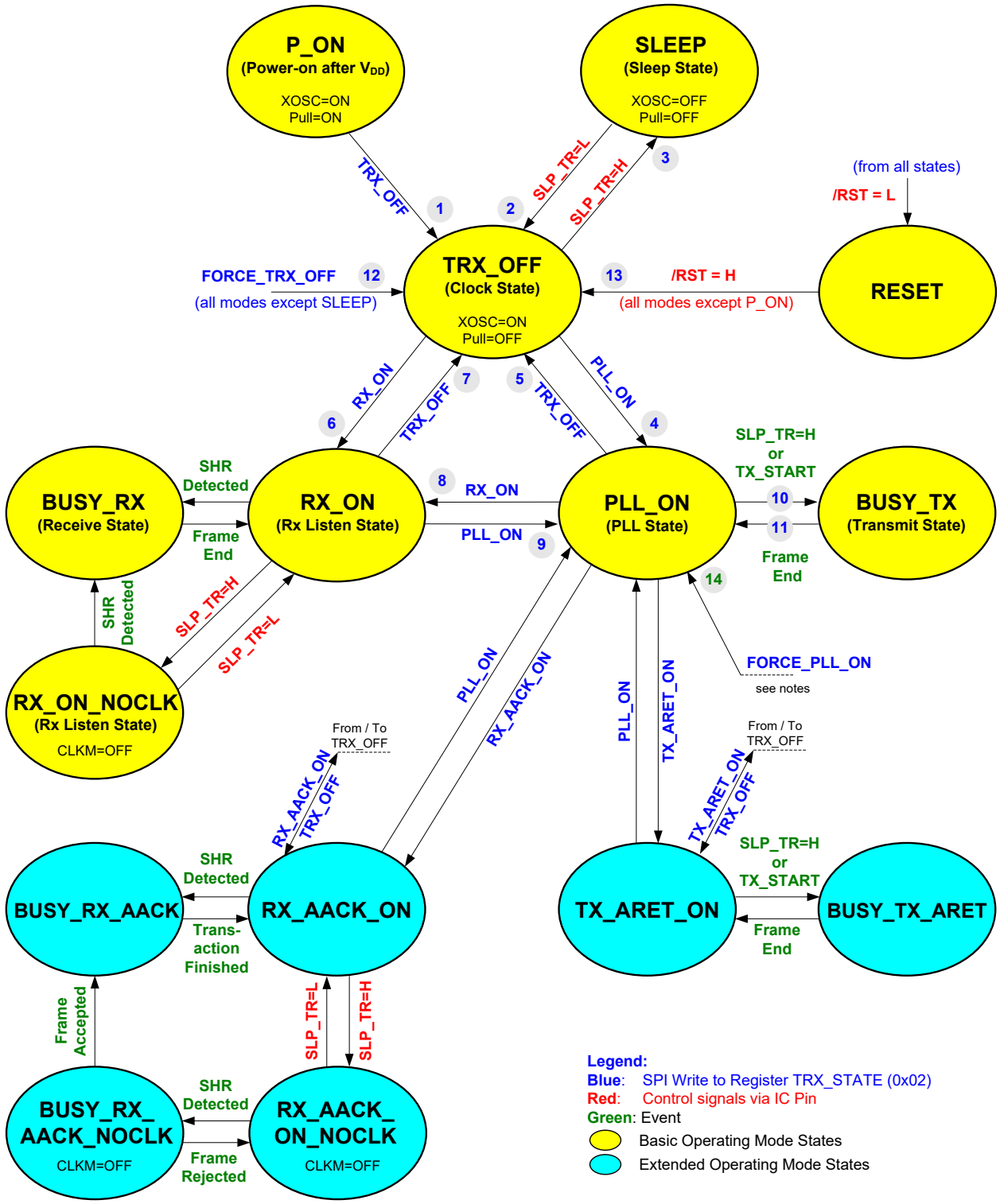
- CSMA-CA, including automatic CCA retry and random backoff
- Frame transmission and automatic FCS field generation
- Reception of ACK frame (if an ACK was requested)
- Automatic retry of transmissions if ACK was expected but not received or accepted
- Interrupt signaling with transaction status

Automatic FCS check and generation, refer to Section 8.3, is used by the RX_AACK and TX_ARET modes. In RX_AACK mode, an automatic FCS check is always performed for incoming frames.

In TX_ARET mode, an ACK which is received within the time required by IEEE 802.15.4 is automatically accepted if the FCS is valid and the ACK sequence number must match the sequence number of the previously transmitted frame. Dependent on the value of the frame pending subfield in the received acknowledgement frame received, the transaction status is set, see the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE.TRAC_STATUS).

An AT86RF212B state diagram, including the Extended Operating Mode states, is shown in the figure below. Orange marked states represent the Basic Operating Mode; blue marked states represent the Extended Operating Mode.

Figure 14-26. Extended Operating Mode State Diagram



Related Links

14.8.2 TRX_STATE

14.2.2.1 State Control

The Extended Operating Mode include RX_AACK and TX_ARET modes and are controlled by writing respective command to register TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). Receive with Auto matic Acknowledgement state RX_AACK_ON and Transmit with Automatic Frame Retransmission and CSMA-CA Retry state TX_ARET_ON can be entered either from TRX_OFF or PLL_ON state as illustrated in Figure 14-26. The completion of each change state command shall always be confirmed by reading the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS).

RX_AACK - Receive with Automatic Acknowledgement

A state transition to RX_AACK_ON from PLL_ON or TRX_OFF is initiated by writing the RX_AACK_ON command to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). On success, reading the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS) returns RX_AACK_ON or BUSY_RX_AACK. The latter one is returned when a frame is being received.

The RX_AACK Extended Operating Mode is terminated by writing command PLL_ON to the register bits TRX_CMD. If the AT86RF212B is within a frame receive or acknowledgment procedure (BUSY_RX_AACK), the state change is executed after finishing. Alternatively, the commands FORCE_TRX_OFF or FORCE_PLL_ON can be used to cancel the RX_AACK transaction and switch to TRX_OFF or PLL_ON state respectively.

TX_ARET - Transmit with Automatic Frame Retransmission and CSMA-CA Retry

A state transition to TX_ARET_ON from PLL_ON or TRX_OFF is initiated by writing TX_ARET_ON command to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). The radio transceiver is in the TX_ARET_ON state when the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS) return TX_ARET_ON. The TX_ARET transaction (frame transmission) is actually started by a rising edge on pin 11 (SLP_TR) or by writing the command TX_START to the TRX_STATE.TRX_CMD bits.

The TX_ARET Extended Operating Mode is terminated by writing the command PLL_ON to the TRX_STATE.TRX_CMD bits. If the AT86RF212B is in the mids of a CSMA-CA transaction, a frame transmission or an acknowledgment procedure (BUSY_TX_ARET), the state change is executed after completing of the operation. Alternatively, the command FORCE_PLL_ON can be used to instantly terminate the TX_ARET transaction and change into transceiver state PLL_ON, respectively.

Note:

1. A state change request from TRX_OFF to RX_AACK_ON or TX_ARET_ON internally passes through PLL_ON state to initiate the radio transceiver front end. Inserting PLL_ON state and associated delays while performing this transition are indicated in. State transitioning can be tracked when interrupt IRQ_0 (PLL_LOCK) is used as an indicator.

Related Links

14.2.2 Extended Operationg Mode
14.2.1.4.6 State Transition Timing Summary
14.8.2 TRX_STATE
14.8.1 TRX_STATUS

14.2.2.2 Configuration

As the usage of the Extended Operating Mode is based on Basic Operating Mode functionality, only features beyond the basic radio transceiver functionality are described in the following sections. For details refer to *Basic Operating Mode*.

When using the RX_AACK or TX_ARET modes, the following registers need to be configured.

RX_AACK configuration steps:

- Set the short address, PAN ID, and IEEE address; SHORT_ADDRESS_x, PAN_ID_x and IEEE_ADDR_x registers
- Configure RX_AACK properties; XAH_CTRL_0 and CSMA_SEED_1 registers
  - Handling of Frame Version Subfield
  - Handling of Pending Data Indicator
  - Characterization as PAN coordinator
  - Handling of Slotted Acknowledgement

- Additional Frame Filtering Properties; XAH_CTRL_1 and CSMA_SEED_1 registers
  - Use of Promiscuous Mode
  - Use of automatic ACK generation
  - Handling of reserved frame types

Refer to the *Frame Filter Configuration* for details. The addresses for the address match algorithm are to be stored in the appropriate address registers. Additional control of the RX_AACK mode is done with the XAH_CTRL_1 and CSMA_SEED_1 registers.

As long as a short address is not set, only broadcast frames and frames matching the full 64-bit IEEE address can be received.

Configuration examples for different device operating modes and handling of various frame types can be found in *Description of RX_AACK Configuration Bits*.

TX_ARET configuration steps:

- Set register bit TX_AUTO_CRC_ON = 1; TRX_CTRL_1 register
- Configure CSMA-CA
  - MAX_FRAME_RETRIES; XAH_CTRL_0 register
  - MAX_CSMA_RETRIES; XAH_CTRL_0 register
  - CSMA_SEED; CSMA_SEED_0 and CSMA_SEED_1 registers
  - MAX_BE, MIN_BE; CSMA_BE register
- Configure CCA

The MAX_FRAME_RETRIES bits in the XAH_CTRL_0 register (XAH_CTRL_0.MAX_FRAME_RETRIES) defines the maximum number of frame retransmissions.

The MAX_CSMA_RETRIES bits in the XAH_CTRL_0 register (XAH_CTRL_0.MAX_CSMA_RETRIES) configure the number of CSMA-CA retries after a busy channel is detected.

The CSMA_SEED_0 and CSAM_SEED_1 bits in the CSMA_SEED_0 and CSMA_SEED_1 registers (CSMA_SEED_0 .CSMA_SEED_0[7:0] and CSMA_SEED_1.CSMA_SEED_1[3:0]) defines a random seed for the backoff-time random-number generator in the AT86RF212B.

The MAX_BE and MIN_BE bits in the CSMA_BE register (CSMA_BE.MAX_BE and CSMA_BE.MIN_BE) set the maximum and minimum CSMA backoff exponent (see [2]), respectively.

Related Links

14.2.1 Operating Modes
14.8.39 CSMA_BE
14.8.37 CSMA_SEED_0
14.8.38 CSMA_SEED_1
14.8.4 TRX_CTRL_1
14.8.36 XAH_CTRL_0
14.8.23 XAH_CTRL_1
14.8.35 IEEE_ADDR
14.8.33 PAN_ID_0
14.8.34 PAN_ID_1
14.8.31 SHORT_ADDR_0
14.8.32 SHORT_ADDR_1

14.2.2.3 RX_AACK_ON-Receive with Automatic ACK

The RX_AACK Extended Operating Mode handles reception and automatic acknowledgement of IEEE 802.15.4 compliant frames.

The general functionality of the RX_AACK procedure is shown in the figure below.

The gray shaded area is the standard flow of an RX_AACK transaction for IEEE 802.15.4 compliant frames. All other procedures are exceptions for specific operating modes or frame formats.

In RX_AACK_ON state, the AT86RF212B listens for incoming frames. After detecting a valid PHR, the radio transceiver changes into BUSY_RX_AACK state and parses the frame content of the MAC header (MHR).

If the content of the MAC addressing fields of the received frame (refer to IEEE 802.15.4 Section 7.2.1) matches one of the configured addresses, dependent on the addressing mode, an address match interrupt IRQ_5 (AMI) is issued. The reference address values are to be stored in the SHORT_ADDR_x, PAN_ID_x and IEEE_ADDR_x registers. Frame filtering is also applied in Basic Operating Mode. However, in Basic Operating Mode, the result of frame filtering or FCS check do not affect the generation of an interrupt IRQ_3 (TRX_END).

Generally, at nodes configured as a normal device or a PAN coordinator, a frame is indicated by interrupt IRQ_3 (TRX_END) if the frame passes the Frame Filter and the FCS is valid. The interrupt is issued after the completion of the frame reception. The microcontroller can then read the frame data. An exception applies if promiscuous mode is enabled. In this case, an interrupt IRQ_3 (TRX_END) is issued for all frames.

During reception AT86RF212B parses bit[5] (ACK Request) of the frame control field of the received data or MAC command frame to check if an acknowledgement (ACK) reply is expected. If the bit is set and if the frame passes the third level of filtering, see IEEE 802.15.4-2006, Section 7.5.6.2, the radio transceiver automatically generates and transmits an ACK frame. The sequence number is copied from the received frame.

The content of the frame pending subfield of the ACK response is set by the AACK_SET_PD bit in teh CSMA_SEED_1 register (CSMA_SEED_1.AACK_SET_PD) when the ACK frame is sent in response to a data request MAC command frame, otherwise this subfield is set to zero.

By default, the acknowledgment frame is transmitted *aTurnaroundTime* (12 symbol periods; see IEEE 802.15.4-2006, Section 6.4.1) after the reception of the last symbol of a data or MAC command frame. Optionally, for non-compliant networks, this delay can be reduced to two symbols by the AACK_ACK_TIME bit in the XAM_CTRL_1 register.

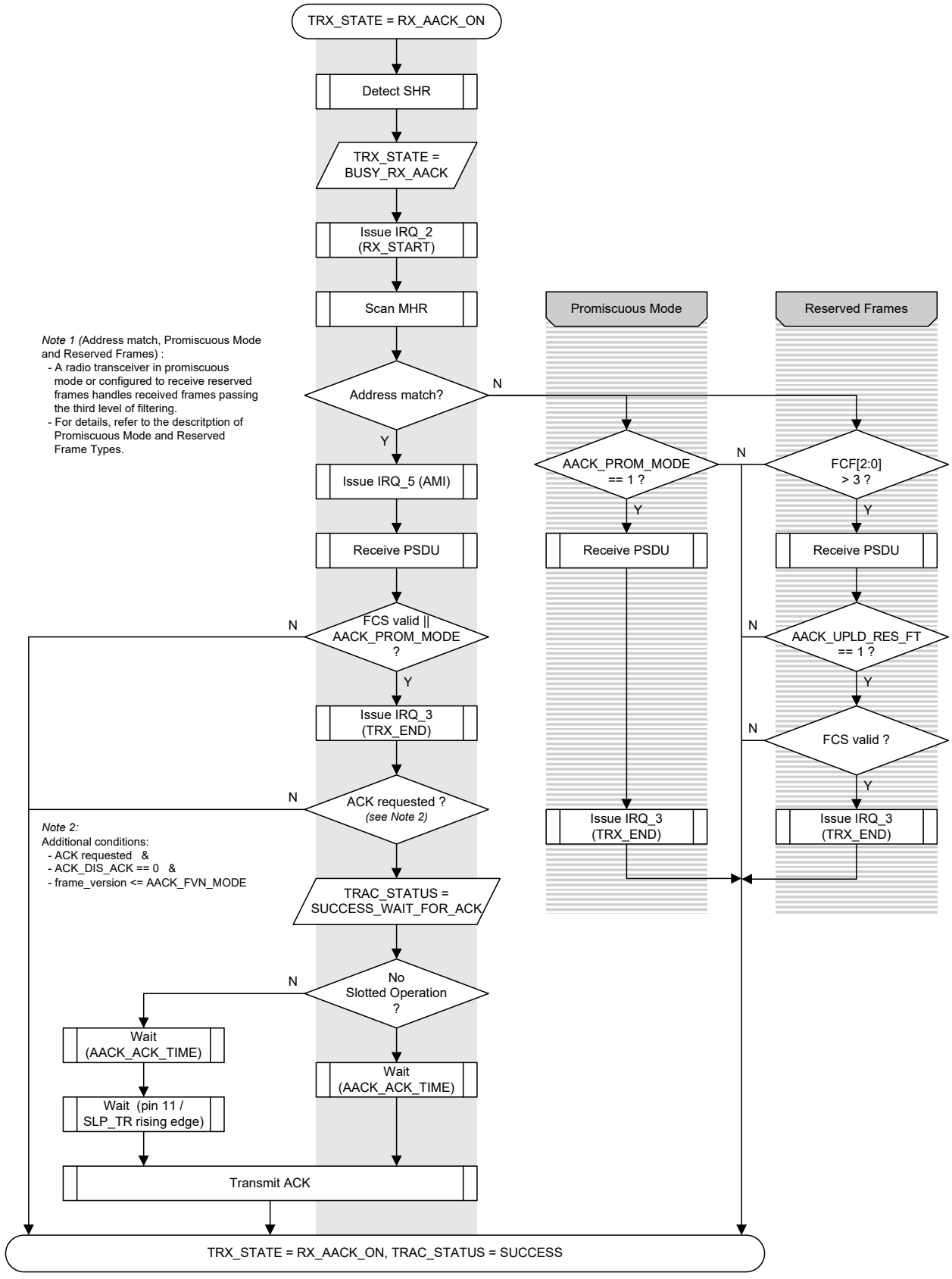
If the AACK_DIS_ACK bit in the CSMA_SEED_1 register (CSMA_SEED_1.AACK_DIS_ACK) is set, no acknowledgement frame is sent even if an acknowledgment frame is requested. This is useful for operating the MAC hardware accelerator in promiscuous mode.

For slotted operation, the start of the transmission of acknowledgement frames is controlled by pin 11 (SLP_TR).

The status of the RX_AACK transaction is indicated by the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE.TRAC_STATUS).

During the operations described above, the AT86RF212B remains in BUSY_RX_AACK state.

Figure 14-27. Flow Diagram of RX_AACK



Related Links

14.8.2 TRX_STATE
14.8.37 CSMA_SEED_0
14.8.38 CSMA_SEED_1
14.8.35 IEEE_ADDR
14.8.33 PAN_ID_0
14.8.34 PAN_ID_1
14.8.31 SHORT_ADDR_0
14.8.32 SHORT_ADDR_1

14.2.2.3.1 Description of RX_AACK Configuration Bits

RX_AACK configuration as described below shall be done prior to switching the AT86RF212B into state RX_AACK_ON.

The table below summarizes all register bits which affect the behavior of an RX_AACK transaction. For frame filtering it is further required to setup address registers to match the expected address.

Table 14-9. Overview of RX_AACK Configuration Bits

Register Address	Register Bits	Register Name	Description
0x20,0x21 0x22,0x23 0x24 … 0x2B		SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7	Setup Frame Filter.
0x0C	7	RX_SAFE_MODE	Dynamic frame buffer protection.
0x17	1	AACK_PROM_MODE	Support promiscuous mode.
0x17	2	AACK_ACK_TIME	Change auto acknowledge start time.
0x17	4	AACK_UPLD_RES_FT	Enable reserved frame type reception, needed to receive non-standard compliant frames.
0x17	5	AACK_FLTR_RES_FT	Filter reserved frame types like data frame type, needed for filtering of non-standard compliant frames.
0x2C	0	SLOTTED_OPERATION	If set, acknowledgment transmission has to be triggered by pin 11 (SLP_TR).
0x2E	3	AACK_I_AM_COORD	If set, the device is a PAN coordinator, that is responds to a null address.
0x2E	4	AACK_DIS_ACK	Disable generation of acknowledgment.
0x2E	5	AACK_SET_PD	Set frame pending subfield in Frame Control Field (FCF).
0x2E	7:6	AACK_FVN_MODE	Controls the ACK behavior, depending on FCF frame version number.

The usage of the RX_AACK configuration bits for various operating modes of a node is explained in the following sections. Configuration bits not mentioned in the following two sections should be set to their reset values.

The general behavior of the *AT86RF212B Extended Feature Set* settings:

- SFD_VALUE (alternative SFD value)
- ANT_DIV (Antenna Diversity)

- RX_PDT_LEVEL (blocking frame reception of lower power signals)

are completely independent from RX_AACK mode and can be arbitrarily combined.

Related Links

14.2.2.1 State Control
14.2.2.3.2 Configuration of IEEE 802.15.4 Compliant Scenarios
14.2.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios
14.7 Register Summary
14.9 Reset Values

14.2.2.3.2 Configuration of IEEE 802.15.4 Compliant Scenarios

Device not operating as a PAN Coordinator

The table below shows a typical AT86RF212B RX_AACK configuration of an IEEE 802.15.4 device operating as a normal device, rather than a PAN coordinator or router.

Table 14-10. Configuration of IEEE 802.15.4 Devices

Register Address	Register Bits	Register Name	Description
0x20,0x21 0x22,0x23 0x24 … 0x2B		SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7	Setup Frame Filter
0x0C	7	RX_SAFE_MODE	*0*: Disable frame protection. *1*: Enable frame protection.
0x2C	0	SLOTTED_OPERATION	*0*: Slotted acknowledgment transmissions are not to be used. *1*: Slotted acknowledgment transmissions are to be used.
0x2E	7:6	AACK_FVN_MODE	Controls the ACK behavior, depending on FCF frame version number. *b00*: Acknowledges only frames with version number 0, that is according to IEEE 802.15.4-2003 frames. *b01*: Acknowledges only frames with version number 0 or 1, that is frames according to IEEE 802.15.4-2006. *b10*: Acknowledges only frames with version number 0 or 1 or 2. *b11*: Acknowledges all frames, independent of the FCF frame version number.

Notes:

1. The default value of the short address is 0xFFFF. Thus, if no short address has been configured, only frames with either the broadcast address or the IEEE address are accepted by the frame filter.
2. In the IEEE 802.15.4-2003 standard the frame version subfield does not yet exist but is marked as reserved. According to this standard, reserved fields have to be set to zero. At the same time, the IEEE 802.15.4-2003 standard requires ignoring reserved bits upon reception. Thus, there is a contradiction in the standard which can be interpreted in two ways:
3. If a network should only allow access to nodes compliant to IEEE 802.15.4-2003, then AACK_FVN_MODE should be set to zero.
4. If a device should acknowledge all frames independent of its frame version, AACK_FVN_MODE should be set to three. However, this may result in conflicts with co-existing IEEE 802.15.4-2006 standard compliant networks.

The same holds for PAN coordinators, see below.

PAN Coordinator

The table below shows the AT86RF212B RX_AACK configuration for a PAN coordinator.

Table 14-11. Configuration of a PAN Coordinator

Register Address	Register Bits	Register Name	Description
0x20,0x21 0x22,0x23 0x24 … 0x2B		SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7	Setup Frame Filter.
0x0C	7	RX_SAFE_MODE	*0*: Disable frame protection. *1*: Enable frame protection.
0x2C	0	SLOTTED_OPERATION	*0*: Slotted acknowledgment transmissions are not to be used. *1*: Slotted acknowledgment transmissions are to be used.
0x2E	3	AACK_I_AM_COORD	*1*: Device is PAN coordinator.
0x2E	5	AACK_SET_PD	*0*: Frame pending subfield is not set in FCF. *1*: Frame pending subfield is set in FCF.
0x2E	7:6	AACK_FVN_MODE	Controls the ACK behavior, depends on FCF frame version number. *b00*: Acknowledges only frames with version number 0, that is according to IEEE 802.15.4-2003 frames. *b01*: Acknowledges only frames with version number 0 or 1, that is frames according to IEEE 802.15.4-2006. *b10*: Acknowledges only frames with version number 0 or 1 or 2. *b11*: Acknowledges all frames, independent of the FCF frame version number.

Promiscuous Mode or Sniffer

The promiscuous mode is described in IEEE 802.15.4-2006, Section 7.5.6.5. This mode is further illustrated in Figure 14-27. According to IEEE 802.15.4-2006 when in promiscuous mode, the MAC sub layer shall pass received frames with correct FCS to the next higher layer and shall not process them further. This implies that received frames should never be automatically acknowledged.

In order to support sniffer application and promiscuous mode, only second level filter rules as defined by IEEE 802.15.4-2006, Section 7.5.6.2, are applied to the received frame.

The table below shows a typical configuration of a device operating in promiscuous mode.

Table 14-12. Configuration of Promiscuous Mode

Register Address	Register Bits	Register Name	Description
0x20,0x21 0x22,0x23 0x24 … 0x2B		SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7	Each address shall be set: 0x00.
0x17	1	AACK_PROM_MODE	*1*: Enable promiscuous mode.
0x2E	4	AACK_DIS_ACK	*1*: Disable generation of acknowledgment.
0x2E	7:6	AACK_FVN_MODE	Controls the ACK behavior, depends on FCF frame version number. *b00*: Acknowledges only frames with version number 0, that is according to IEEE 802.15.4-2003 frames. *b01*: Acknowledges only frames with version number 0 or 1, that is frames according to IEEE 802.15.4-2006. *b10*: Acknowledges only frames with version number 0 or 1 or 2. *b11*: Acknowledges all frames, independent of the FCF frame version number.

If the AT86RF212B radio transceiver is in promiscuous mode, second level of filtering according to IEEE 802.15.4-2006, Section 7.5.6.2, is applied to a received frame. However, an IRQ_3 (TRX_END) is issued even if the FCS is invalid. Thus, it is necessary to read the RX_CRC_VALID bit in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID) after IRQ_3 (TRX_END) in order to verify the reception of a frame with a valid FCS. Alternatively, bit[7] of byte RX_STATUS can be evaluated.

If a device, operating in promiscuous mode, receives a frame with a valid FCS which further passed the third level of filtering according to IEEE 802.15.4-2006, Section 7.5.6.2, and an acknowledgement (ACK) frame would be transmitted. But, according to the definition of the promiscuous mode, a received frame shall not be acknowledged, even if requested. Thus, the AACK_DIS_ACK bit in the CSMA_SEED_1 register (CSMA_SEED_1.AACK_DIS_ACK) must be set to one to disable ACK generation.

In all receive modes IRQ_5 (AMI) interrupt is issued, when the received frame matches the node's address according to the filter rules described in the *Frame Filter* section

Alternatively, in state RX_ON (Basic Operating Mode), when a valid PHR is detected, an IRQ_2 (RX_START) is generated and the frame is received. The end of the frame reception is signalized with an IRQ_3 (TRX_END). At the same time the RX_CRC_VALID bit in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID) is updated with the result of the FCS check. According to the promiscuous mode definition the register bit RX_CRC_VALID needs to be checked in order to dismiss corrupted frames.

However, the RX_AACK transaction additionally enables extended functionality like automatic acknowledgement and non-destructive frame filtering.

Related Links

14.1.3.2 Frame Buffer Access Mode
14.2.1 Operating Modes
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.2.2.3.5 RX_AACK Slotted Operation – Slotted Acknowledgement
14.8.38 CSMA_SEED_1
14.8.6 PHY_RSSI

14.2.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios

Sniffer

The table below shows an AT86RF212B RX_AACK configuration to setup a sniffer device. Other RX_AACK configuration bits, refer to Table 14-9, should be set to their reset values.

All frames received are indicated by an IRQ_2 (RX_START) and IRQ_3 (TRX_END). After frame reception the RX_CRC_VALID bit in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID) is updated with the result of the FCS check. The RX_CRC_VALID bit needs to be checked in order to dismiss corrupted frames.

Table 14-13. Configuration of a Sniffer Device

Register Address	Register Bits	Register Name	Description
0x17	1	AACK_PROM_MODE	*1*: Enable promiscuous mode.
0x2E	4	AACK_DIS_ACK	*1*: Disable generation of acknowledgment.

This operating mode is similar to the promiscuous mode.

Reception of Reserved Frames

In RX_AACK mode, frames with reserved frame types can also be handled. This might be required when implementing proprietary, non-standard compliant, protocols. The reception of reserved frame types is an extension of the AT86RF212B Frame Filter. Received frames are either handled like data frames, or may be allowed to completely bypass the Frame Filter. The flow chart in Figure 14-27 shows the corresponding state machine.

In addition to Table 14-10 or Table 14-11, the table below shows RX_AACK configuration registers required to setup a node to receive reserved frame types.

Table 14-14. RX_AACK Configuration to Receive Reserved Frame Types

Register Address	Register Bits	Register Name	Description
0x17	4	AACK_UPLD_RES_FT	*1*: Enable reserved frame type reception.
0x17	5	AACK_FLTR_RES_FT	Filter reserved frame types like data frame type, see note below. *0*: Disable reserved frame types filtering. *1*: Enable reserved frame types filtering.

There are three different options for handling reserved frame types.

1. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 0:
   Any non-corrupted frame with a reserved frame type is indicated by an IRQ_3 (TRX_END) interrupt. No further address filtering is applied on those frames. An IRQ_5 (AMI) interrupt is never generated and the acknowledgment subfield is ignored.
2. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 1:

If AACK_FLT_RES_FT = 1 any frame with a reserved frame type is filtered by the address filter similar to a data frame as described in the standard. This implies the generation of the IRQ_5 (AMI) interrupts upon address match. An IRQ_3 (TRX_END) interrupt is only generated if the address matched and the frame was not corrupted. An acknowledgment is only send, when the ACK request subfield was set in the received frame and an IRQ_3 (TRX_END) interrupt occurred.

3. AACK_UPLD_RES_FT = 0:
   Any received frame with a reserved frame type is discarded.

Short Acknowledgment Frame (ACK) Start Timing

The AACK_ACK_TIME bit in the XAH_CTRL_1 register (XAH_CTRL_1.AACK_ACK_TIME), defines the delay between the end of the frame reception and the start of the transmission of an acknowledgment frame.

Table 14-15. ACK Start Timing for Unslotted Operation

Register Address	Register Bit	Register Name	Description
0x17	2	AACK_ACK_TIME	*0*: IEEE 802.15.4 standard compliant acknowledgement timing of 12 symbol periods. *1*: Non-standard IEEE 802.15.4 reduced acknowledgment delay is set to two symbol periods (BPSK-20, O-QPSK-{100,200,400}) or three symbol periods (BPSK-40, O-QPSK-{250,500,1000}).

This feature can be used in all scenarios, independent of other configurations. However, shorter acknowledgment timing is especially useful when using High Data Rate Modes to increase battery lifetime and to improve the overall data throughput;.

In slotted operation mode, the acknowledgment transmission is actually started by pin 11 (SLP_TR). The table below shows that the AT86RF212B enables the trigger pin with an appropriate delay. Thus, a transmission cannot be started earlier.

Table 14-16. ACK Start Timing for Slotted Operation

Register Address	Register Bit	Register Name	Description
0x17	2	AACK_ACK_TIME	*0*: Acknowledgment frame transmission can be triggered after six symbol periods. *1*: Acknowledgment frame transmission can be triggered after three symbol periods.

Related Links

14.2.2.3.1 Description of RX_AACK Configuration Bits
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.8.23 XAH_CTRL_1
14.8.6 PHY_RSSI

14.2.2.3.4 RX_AACK_NOCLK – RX_AACK_ON without CLKM

If the AT86RF212B is listening for an incoming frame and the microcontroller is not running an application, the microcontroller can be powered down to decrease the total system power consumption. This special power-down scenario for systems running in clock synchronous mode is supported by the AT86RF212B using the states RX_AACK_ON_NOCLK and BUSY_RX_AACK_NOCLK, see Figure 14-26. They achieve the same functionality as the states RX_AACK_ON and BUSY_RX_AACK with pin 17 (CLKM) disabled.

The RX_AACK_NOCLK state is entered from RX_AACK_ON by a rising edge at pin 11 (SLP_TR). The return to RX_AACK_ON state automatically results either from the reception of a valid frame, indicated by interrupt IRQ_3 (TRX_END), or a falling edge on pin 11 (SLP_TR).

A received frame is considered valid if it passes frame filtering and has a correct FCS. If an ACK was requested, the radio transceiver enters BUSY_RX_AACK state and follows the procedure described in the *RX_AACK_ON – Receive with Automatic ACK* section.

After the RX_AACK transaction has been completed, the radio transceiver remains in RX_AACK_ON state. The AT86RF212B re-enters the RX_AACK_ON_NOCLK state only by the next rising edge on pin 11 (SLP_TR).

The timing and behavior, when CLKM is disabled or enabled, are described in the *Sleep/Wake-up and Transmit Signal (SLP_TR)* section.

Note: RX_AACK_NOCLK is not available for slotted operation mode.

Related Links

14.1.6 Sleep/Wake-up and Transmit Signal (SLP_TR)
14.2.2 Extended Operationg Mode
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.2.2.3.5 RX_AACK Slotted Operation – Slotted Acknowledgement

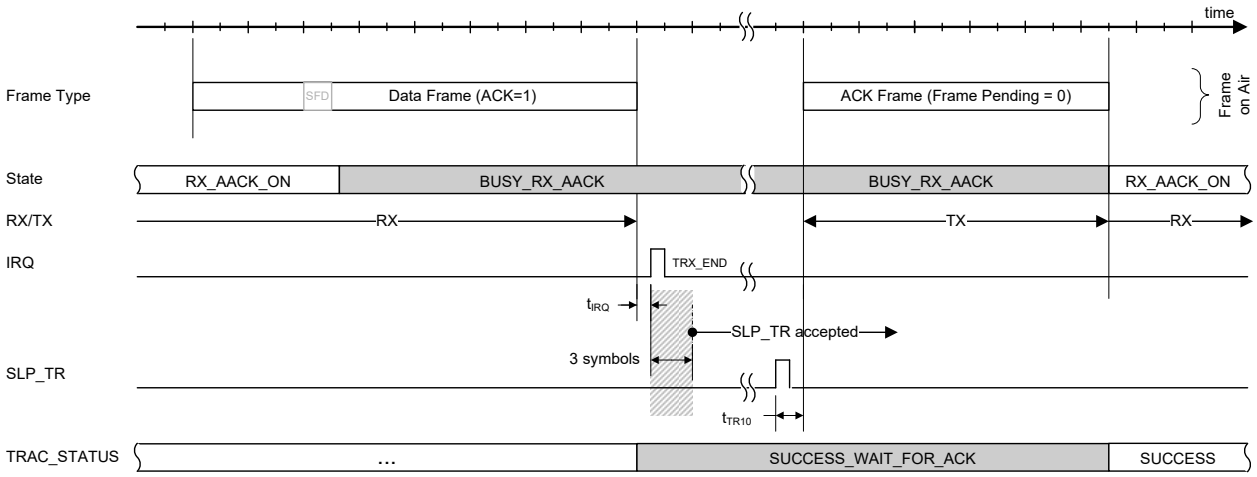
14.2.2.3.5 RX_AACK Slotted Operation – Slotted Acknowledgement

In networks using slotted operation the start of the acknowledgment frame, and thus the exact timing, must be provided by the microcontroller. Exact timing requirements for the transmission of acknowledgments in beacon-enabled networks are explained in IEEE 802.15.4-2006, Section 7.5.6.4.2. In conjunction with the microcontroller the AT86RF212B supports slotted acknowledgement operation. This mode is invoked by setting the SLOTTED_OPERATION bit in the XAH_CTRL_0 register (XAH_CTRL_0.SLOTTED_OPERATION) to one.

If an acknowledgment (ACK) frame is to be transmitted in RX_AACK mode, the radio transceiver expects a rising edge on pin 11 (SLP_TR) to actually start the transmission. During this waiting period, the transceiver reports SUCCESS_WAIT_FOR_ACK through the TRAC_STATUS bits in the XAH_CTRL_0 register (XAH_CTRL_0.TRAC_STATUS), see Figure 14-27. The minimum delay between the occurrence of interrupt IRQ_3 (TRX_END) and pin start of the ACK frame in slotted operation is three symbol periods.

The figure below illustrates the timing of an RX_AACK transaction in slotted operation. The acknowledgement frame is ready to transmit three symbol times after the reception of the last symbol of a data or MAC command frame indicated by IRQ_3 (TRX_END). The transmission of the acknowledgement frame is initiated by the microcontroller with the rising edge of pin 11 (SLP_TR) and starts t_{TR10} = 1symbol period later. The interrupt latency t_{IRQ} is specified in the *Digital Interface Timing Characteristics* section.

Figure 14-28. Timing Example of an RX_AACK Transaction for Slotted Operation



Related Links

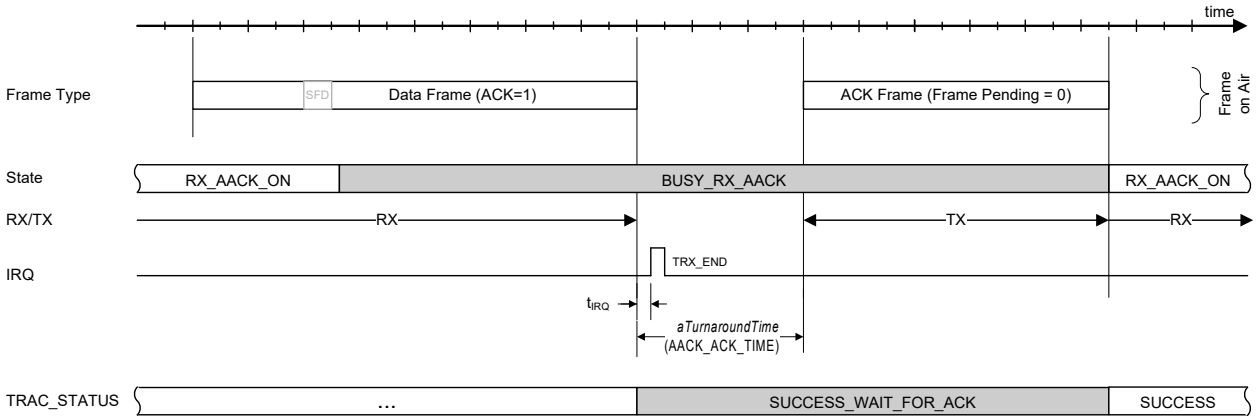
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.8.36 XAH_CTRL_0

14.2.2.3.6 RX_AACK Mode Timing

A timing example of an RX_AACK transaction is shown in the figure below. In this example, a data frame with an ACK request is received. The AT86RF212B changes to state BUSY_RX_AACK after SFD detection. The completion

of the frame reception is indicated by an IRQ_3 (TRX_END) interrupt. The interrupts IRQ_2 (RX_START) and IRQ_5 (AMI) are disabled in this example. The ACK frame is automatically transmitted after *aTurnaroundTime* (12 symbols), assuming default acknowledgment frame start timing. The interrupt latency t_{IRQ} is specified in the *Digital Interface Timing Characteristics*section.

Figure 14-29. Timing Example of an RX_AACK Transaction



Note:

1. If the AACK_ACK_TIME bit in the XAH_CTRL_1 register (XAH_CTRL_1.AACK_ACK_TIME) is set, an acknowledgment frame is sent already two or three symbol times after the reception of the last symbol of a data or MAC command frame.

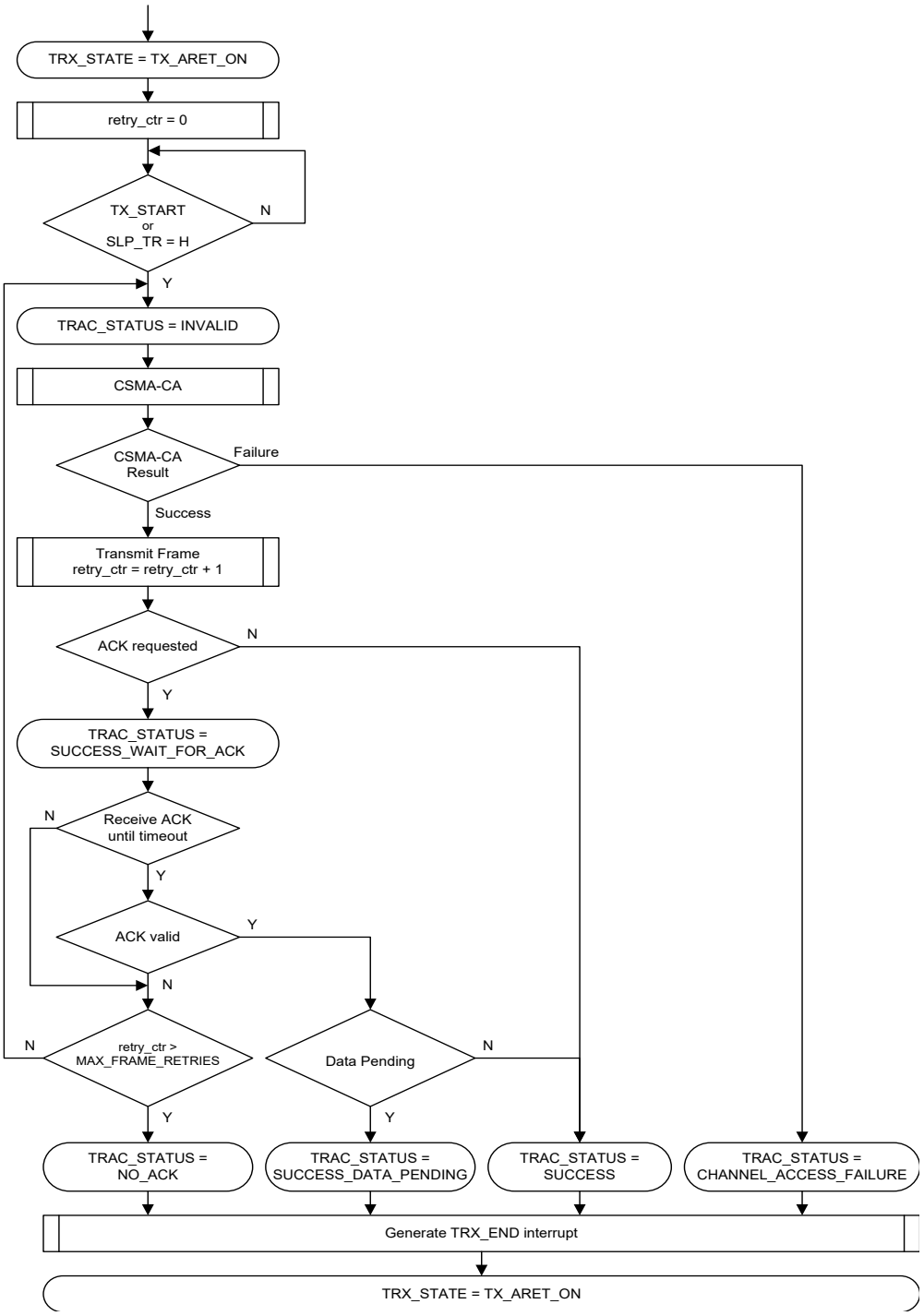
Related Links

14.2.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios

14.8.23 XAH_CTRL_1

14.2.2.4 TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry

Figure 14-30. Flow Diagram of TX_ARET



Overview

The implementation of TX_ARET algorithm is shown in the figure above.

The TX_ARET Extended Operating Mode supports the frame transmission process as defined by IEEE 802.15.4-2006. It is invoked by writing TX_ARET_ON to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD).

If a transmission is initiated in TX_ARET mode, the AT86RF212B executes the CSMA-CA algorithm as defined by IEEE 802.15.4-2006, Section 7.5.1.4. If the CCA reports IDLE, the frame is transmitted from the Frame Buffer.

If an acknowledgement frame is requested, the radio transceiver checks for an ACK reply automatically. The CSMA-CA based transmission process is repeated until a valid acknowledgement is received or the number of frame retransmissions is exceeded, refer to the MAX_FRAME_RETRIES bits in the XAH_CTRL_0 register..

The completion of the TX_ARET transaction is indicated by the IRQ_3 (TRX_END) interrupt.

Description

Prior to invoking AT86RF212B TX_ARET mode, the basic configuration steps must be executed. It is further recommended to write the PSDU transmit data to the Frame Buffer in advance.

The transmit start event may either come from a rising edge on pin 11 (SLP_TR) or by writing a TX_START command to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD).

If the CSMA-CA detects a busy channel, it is retried as specified by the MAX_CSMA_RETRIES bits in the XAH_CTRL_0 register (XAH_CTRL_0.MAX_CSMA_RETRIES). In case that CSMA-CA does not detect a clear channel after MAX_CSMA_RETRIES, it aborts the TX_ARET transaction, issues interrupt IRQ_3 (TRX_END), and sets the value of the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE_TRAC.STATUS) to CHANNEL_ACCESS_FAILURE.

During transmission of a frame the radio transceiver parses bit[5] (ACK Request) of the MAC header (MHR) frame control field of the PSDU data (PSDU octet #1) to be transmitted to check if an ACK reply is expected.

If no ACK is expected, the radio transceiver issues IRQ_3 (TRX_END) directly after the frame transmission has been completed. The TRX_STATE.TRAC_STATUS bits are set to SUCCESS.

If an ACK is expected, after transmission the radio transceiver automatically switches to receive mode waiting for a valid ACK reply (that is matching sequence number and correct FCS). After receiving a valid ACK frame, the “Frame Pending” subfield of this frame is parsed and the TRX_STATE.TRAC_STATUS bits re updated to SUCCESS or SUCCESS_DATA_PENDING accordingly. At the same time, the entire TX_ARET transaction is terminated and interrupt IRQ_3 (TRX_END) is issued.

If no valid ACK is received within the timeout period, the radio transceiver retries the entire transaction (CSMA-CA based frame transmission) until the maximum number of frame retransmissions is exceeded, see the XAH_CTRL_0.MAX_FRAME_RETRIES bits. In that case, the TRX_STATE.TRAC_STATUS is set to NO_ACK, the TX_ARET transaction is terminated, and interrupt IRQ_3 (TRX_END) is issued.

Note:

1. The acknowledgment receive procedure does not overwrite the Frame Buffer content. Transmit data in the Frame Buffer is not modified during the entire TX_ARET transaction. Received frames, other than the expected ACK frame, are discarded automatically.

After that, the microcontroller may read the value of the TRX_STATE.TRAC_STATUS bits to verify whether the transaction was successful or not. The register bits are set according to the following cases.

The table below summarizes the Extended Operating Mode result codes in the TRX_STATE.TRAC_STATUS bits with respect to the TX_ARET transaction. Values are meaningful after an interrupt until the next frame transmit.

Table 14-17. Interpretation of TRAC_STATUS Register Bits

Value	Name	Description
0	SUCCESS	The transaction was responded to by a valid ACK, or, if no ACK is requested, after a successful frame transmission.
1	SUCCESS_DATA_PENDING	Equivalent to SUCCESS and indicating that the “Frame Pending” bit (see Section 8.1.2.2) of the received acknowledgment frame was set.
3	CHANNEL_ACCESS_FAILURE	Channel is still busy after attempting MAX_CSMA_RETRIES of CSMA-CA.
5	NO_ACK	No acknowledgement frame was received during all retry attempts.
7	INVALID	

A value of MAX_CSMA_RETRIES = 7 initiates an immediate TX_ARET transaction without performing CSMA-CA. This can be used for example to transmit indirect data to a device. Further the value MAX_FRAME_RETRIES is ignored and the TX_ARET transaction is performed only once.

Related Links

14.1.6 Sleep/Wake-up and Transmit Signal (SLP_TR)
14.2.2.1 State Control
14.2.2.5 Interrupt Handling
14.2.2.2 Configuration
14.2.2.4.1 Acknowledgment Timeout
14.8.2 TRX_STATE
14.8.36 XAH_CTRL_0

14.2.2.4.1 Acknowledgment Timeout

If an acknowledgment (ACK) frame is expected following the frame transmission, the AT86RF212B sets a timeout for the ACK frame to arrive. This timeout *macAckWaitDuration* isdefined according to [2] as follows:

macAckWaitDuration [symbol periods] =

aUnitBackoffPeriod + aTurnaroundTime + phySHRDuration + 6 x phySymbolsPerOctet

where six represents the number of PHY header octets plus the number of PSDU octets in an acknowledgment frame.

Specifically for the implemented PHY Modes, this formula results in the following values:

- BPSK: *macAckWaitDuration* = 120 symbol periods
- O-QPSK: *macAckWaitDuration* = 54 symbol periods

Note:

1. For any PHY Mode the unit “symbol period” refers to the symbol duration of the appropriate synchronization header; refer to the *Symbol Period* section for further information regarding symbol period.

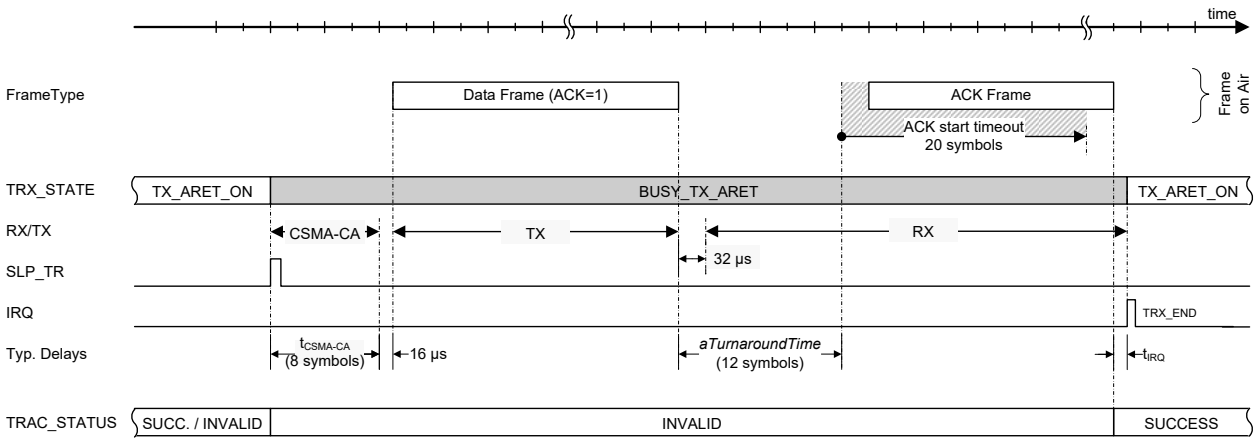
14.2.2.4.2 Timing

A timing example of a TX_ARET transaction is shown in figure below. In the example shown, a data frame with an acknowledgment request is to be transmitted. The frame transmission is started by sending a pulse on pin 11 (SLP_TR). By setting the MIN_BE bits in the CSMA_BE register (CSMA_BE.MIN_BE) to zero, the initial CSMA-CA backoff period is configured to zero length. Thus, the CSMA-CA duration time $t_{CSMA-CA}$ consists only of eight symbols of CCA measurement period. If CCA returns IDLE (assumed here), the frame is transmitted.

Upon frame transmission AT86RF212B switches to the receive mode and expects an acknowledgement response. This is indicated by the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE.TRAC_STATUS) set to SUCCESS_WAIT_FOR_ACK. After a period of *aTurnaroundTime* + *aUnitBackoff*, the transmission of the ACK frame must be started. During the entire transaction, including frame transmit, wait for ACK, and ACK receive, the radio transceiver TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS) are set to BUSY_TX_ARET state.

A successful reception of the acknowledgment frame is indicated by triggering of IRQ_3 (TRX_END). The TRX_STATUS.TRX_STATUS bits changes back to TX_ARET_ON state. When the frame pending subfield of the received ACK frame is set to one (more data is to follow) the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE.TRAC_STATUS) are set either to SUCCESS_DATA_PENDING status instead of SUCCESS status.

Figure 14-31. Timing Example of a TX_ARET Transaction (without Pending Data Bit set in ACK Frame)



Register settings: 0x2C: MAX_FRAME_RETRIES=0 0x2C: MAX_CSMA_RETRIES=0 0x2E: MIN_BE=0

Related Links

14.8.39 CSMA_BE
14.8.2 TRX_STATE
14.8.1 TRX_STATUS

14.2.2.5 Interrupt Handling

The AT86RF212B interrupt handling in the Extended Operating Mode is similar to the Basic Operating Mode. Interrupts can be enabled by setting the appropriate bit in the IRQ_MASK register.

For RX_AACK and TX_ARET modes the following interrupts inform about the status of a frame reception and transmission:

Table 14-18. Interrupt Handling in Extended Operating Mode

Mode	Interrupt	Description
RX_AACK	IRQ_2 (RX_START)	Indicates a PHR reception
	IRQ_5 (AMI)	Issued at address match
	IRQ_3 (TRX_END)	Signals completion of RX_AACK transaction if successful • A received frame must pass the address filter • The FCS is valid
TX_ARET	IRQ_3 (TRX_END)	Signals completion of TX_ARET transaction
RX_AACK/ TX_ARET	IRQ_0 (PLL_LOCK)	Entering RX_AACK_ON or TX_ARET_ON state from TRX_OFF state, the PLL_LOCK interrupt signals that the transaction can be started

RX_AACK

For support of the RX_AACK functionality, it is recommended to enable IRQ_3 (TRX_END). This interrupt is issued only if frames pass the frame filtering, and have a valid FCS to reflect data validity. This functionality differs in Basic Operating Mode. The usage of other interrupts is optional.

On reception of a valid PHR an IRQ_2 (RX_START) is issued. IRQ_5 (AMI) indicates address match, refer to the rules in the *filter filter*, and the completion of a frame reception with a valid FCS is indicated by interrupt IRQ_3 (TRX_END).

Thus, it can happen that an IRQ_2 (RX_START) and/or IRQ_5 (AMI) are issued, but the IRQ_3 (TRX_END) interrupt is never triggered when a frame does not pass the FCS computation check.

TX_ARET

The IRQ_3 (TRX_END) interrupt is always generated after completing a TX_ARET transaction. Subsequently the transaction status can be read from the TRAC_STATUS bits in the TRX_STATE register (TRX_STATE.TRAC_STATUS).

Several interrupts are automatically suppressed by the radio transceiver during TX_ARET transaction. The CCA algorithm (part of CSMA-CA) does not generate interrupt IRQ_4 (CCA_ED_DONE). Furthermore, the interrupts IRQ_2 (RX_START) and/or IRQ_5 (AMI) are not generated during the TX_ARET acknowledgment receive process.

All other interrupts as described in Section 6.7, are also available in Extended Operating Mode.

Related Links

14.1.7 Interrupt Logic
14.3.2 Frame Filter
14.3.6 Clear Channel Assessment
14.2.1.3 Interrupt Handling
14.8.14 IRQ_MASK

14.3 Functional Description

14.3.1 Introduction – IEEE 802.15.4-2006 Frame Format

The following two figures provides an overview of the physical layer (PHY) frame structure as defined by the IEEE 802.15.4-2006 standard and the medium access control layer (MAC) frame structure.

Related Links

14.3.1.2 MAC Protocol Data Unit (MPDU)
14.3.1.1 PHY Protocol Data Unit (PPDU)

14.3.1.1 PHY Protocol Data Unit (PPDU)

Figure 14-32. IEEE 802.15.4 Frame Format - PHY-Layer Frame Structure (PPDU)

<table>
<tr><td colspan="4">PHY Protocol Data Unit (PPDU)</td></tr>
<tr><td>Preamble Sequence</td><td>SFD</td><td>Frame Length / Reserved</td><td>PHY Service Data Unit (PSDU)</td></tr>
<tr><td colspan="2">5 octets
Synchronization Header (SHR)</td><td>1 octet PHY Header (PHR)</td><td>max. 127 octets
PHY Payload</td></tr>
<tr><td colspan="3"></td><td>MAC Protocol Data Unit (MPDU)</td></tr>
</table>

14.3.1.1.1 Synchronization Header (SHR)

The SHR consists of a four-octet preamble field (all zero), followed by a single byte start-of-frame delimiter (SFD) which has the predefined value 0xA7. During transmission, the SHR is automatically generated by the AT86RF212B, thus the Frame Buffer shall contain PHR and PSDU only.

The transmission of the SHR requires 40 symbols for a transmission with BPSK modulation and 10 symbols for a transmission with O-QPSK modulation. The SHR duration depending on the selected data rate is illustrated in the *Timing Summary*.

The fact that the SPI data rate is normally higher than over-the-air data rate, allows the microcontroller to first initiate a frame transmission and then as the SHR is transmitted write the frame data. This is to minimize frame buffer data fill overhead transmission delay.

During a frame reception, the SHR is used for synchronization purposes. The matching SFD determines the beginning of the PHR and the following PSDU payload data.

Related Links

14.1.3.2 Frame Buffer Access Mode
14.3.1.1.4 Timing Summary
15.15.2 General RF Specifications

14.3.1.1.2 PHY Header (PHR)

The PHY header is a single octet following the SHR. The least significant seven bits denote the frame length of the following PSDU, while the most significant bit of that octet is reserved, and shall be set to zero for IEEE 802.15.4 compliant frames.

On reception, the PHR is returned as the first octet during Frame Buffer read access. While the IEEE 802.15.4-2006 standard declares bit seven of the PHR octet as being reserved, the AT86RF212B preserves this bit upon transmission and reception so it can be used to carry additional information within proprietary networks. Nevertheless, this bit is not considered to be a part of the frame length, so only frames between one and 127 octets are possible. For IEEE 802.15.4 compliant operation bit[7] has to be masked by software.

In transmit mode, the PHR needs to be supplied as the first octet during Frame Buffer write access.

In receive mode, the PHR (that is frame length greater than zero) is returned as the first octet during Frame Buffer read access and is signaled by an interrupt IRQ_2 (RX_START).

Related Links

14.1.3.2 Frame Buffer Access Mode
23. References

14.3.1.1.3 PHY Payload (PHY Service Data Unit, PSDU)

The PSDU has a variable length between zero and *aMaxPHYPacketSize* (127, maximum PSDU size in octets). The length of the PSDU is signaled by the frame length field (PHR), refer to Table 8-1. The PSDU contains the MAC protocol data unit (MPDU), where the last two octets are used for the Frame Check Sequence (FCS).

Received frames with a frame length field set to zero (invalid PHR) are not signaled to the microcontroller.

The table below summarizes the type of payload versus the frame length value.

Table 14-19. Frame Length Field – PHR

Frame Length Value	Payload
0 - 4	Reserved
5	MPDU (Acknowledgement)
6 – 8	Reserved
9 - *aMaxPHYPacketSize*	MPDU

Related Links

14.3.3 Frame Check Sequence (FCS)

14.3.1.1.4 Timing Summary

The table belwo shows timing information for the above mentioned frame structure depending on the selected data rate.

Table 14-20. PPDU Timing

PHY Mode	PSDU Bit Rate [kb/s]	Header Bit Rate [kb/s]	Duration		
			SHR [µs]	PHR [µs]	Max. PSDU [ms]
BPSK(1)	20	20	2000	400	50.8
	40	40	1000	200	25.4
O-QPSK(1)	100	100	300	80	10.16
	250	250	160	32	4.064

.....continued					
PHY Mode	PSDU Bit Rate [kb/s]	Header Bit Rate [kb/s]	Duration		
			SHR [µs]	PHR [µs]	Max. PSDU [ms]
O-QPSK(2)	200	100	300	80	5.08
	400	100	300	80	2.54
	500	250	160	32	2.032
	1000	250	160	32	1.016

Notes:

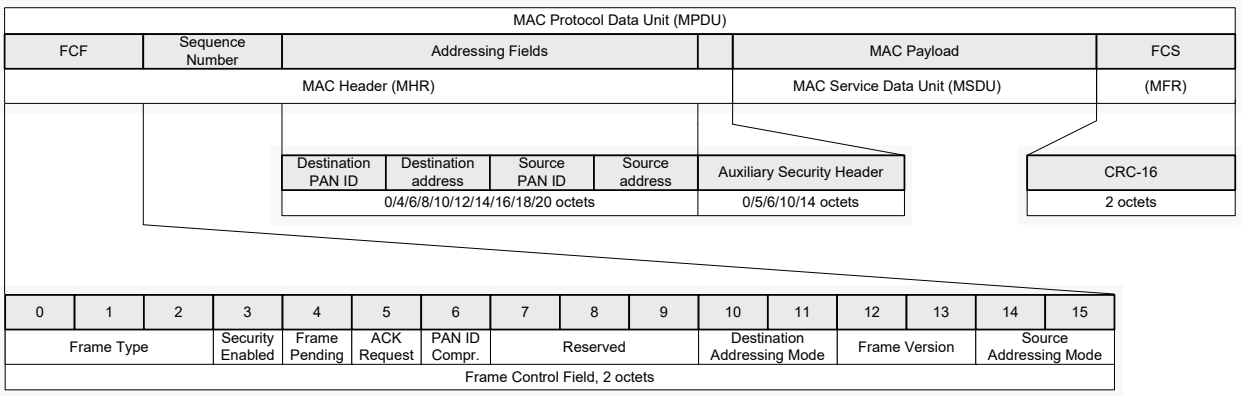
1. Compliant to IEEE 802.15.4-2006 [2].
2. High Data Rate Modes.

Related Links

14.4.1.4 Proprietary High Data Rate Modes

14.3.1.2 MAC Protocol Data Unit (MPDU)

Figure 14-33. IEEE 802.15.4-2006 Frame Format – MAC Layer Frame Structure (MPDU)



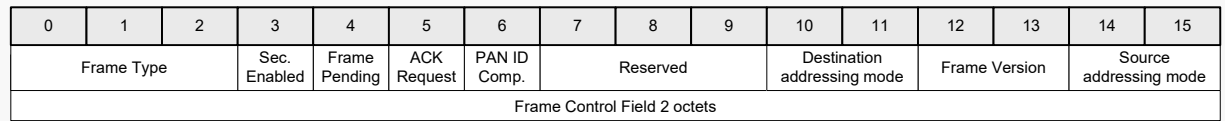
14.3.1.2.1 MAC Header (MHR) Fields

The MAC header consists of the Frame Control Field (FCF), a sequence number, and the addressing fields (which are of variable length, and can even be empty in certain situations).

14.3.1.2.2 Frame Control Field (FCF)

The FCF consists of 16 bits, and occupies the first two octets of the MPDU or PSDU, respectively.

Figure 14-34. IEEE 802.15.4-2006 Frame Control Field (FCF)



Bits [2:0]: describes the “Frame Type“. The table below summarizes frame types defined by IEEE 802.15.4-2006 [2], Section 7.2.1.1.1.

Table 14-21. Frame Control Field – Frame Type Subfield

Frame Control Field Bit Assignments		Description
Frame Type Value b2 b1 b0	Value	
000	0	Beacon
001	1	Data
010	2	Acknowledge
011	3	MAC command
100 – 111	4 – 7	Reserved

This subfield is used for frame filtering by the third level filter rules. By default, only frame types 0 – 3 pass the third level filter rules, refer to Section 8.2. Automatic frame filtering by the AT86RF212B is enabled when using the RX_AACK mode.

However, a reserved frame (frame type value > 3) can be received if the AACK_UPLD_RES_FT bit in the XAH_CTRL_1 register (XAH_CTRL_1.AACK_UPLD_RES_FT) is set.

Frame filtering is also provided in Basic Operating Mode.

Bit 3: indicates whether security processing applies to this frame. This field is evaluated by the Frame Filter.

Bit 4: is the “Frame Pending” subfield. This field can be set in an acknowledgment frame (ACK) in response to a data request MAC command frame. This bit indicates that the node, which transmitted the ACK, might have more data to send to the node receiving the ACK.

Note:

1. For acknowledgment frames automatically generated by the AT86RF212B, this bit is set according to the content of register bit AACK_SET_PD in register 0x2E (CSMA_SEED_1) if the received frame was a data request MAC command frame.

Bit 5: forms the “Acknowledgment Request” subfield. If this bit is set within a data or MAC command frame that is not broadcast, the recipient shall acknowledge the reception of the frame within the time specified by IEEE 802.15.4, that is within 12 symbol periods for nonbeacon-enabled networks.

The radio transceiver parses this bit during RX_AACK mode and transmits an acknowledgment frame if necessary.

In TX_ARET mode this bit indicates if an acknowledgement frame is expected after transmitting a frame. If this is the case, the receiver waits for the acknowledgment frame, otherwise the TX_ARET transaction is finished.

Bit 6: the “PAN ID Compression” subfield, indicates that in a frame where both the destination and source addresses are present, the PAN ID is omitted from the source addressing field. This bit is evaluated by the Frame Filter of the AT86RF212B. This subfield was previously named “Intra-PAN”.

Bits [11:10]: the “Destination Addressing Mode” subfield describes the format of the destination address of the frame. The values of the address modes are summarized in the table below, according to IEEE 802.15.4.

Table 14-22. Frame Control Field – Destination and Source Addressing Mode

Frame Control Field Bit Assignments		Description
Addressing Mode b11 b10 b15 b14	Value	
00	0	PAN identifier and address fields are not present
01	1	Reserved

.....continued

Frame Control Field Bit Assignments		Description
Addressing Mode b11 b10 b15 b14	Value	
10	2	Address field contains a 16-bit short address
11	3	Address field contains a 64-bit extended address

If the destination address mode is either two or three (that is if the destination address is present), it always consists of a 16-bit PAN-ID first, followed by either the 16-bit or 64-bit address as described by the mode.

Bits [13:12]: the “Frame Version” subfield specifies the version number corresponding to the frame. These bits are reserved in IEEE 802.15.4-2003.

This subfield shall be set to zero to indicate a frame compatible with IEEE 802.15.4-2003 and one to indicate an IEEE 802.15.4-2006 frame All other subfield values shall be reserved for future use.

The AACK_FVN_MODE bit in the CSMA_SEED_1 register (CSMA_SEED_1.AACK_FVN_MODE) controls the behavior of frame acknowledgements. This register determines if, depending on the Frame Version Number, a frame is acknowledged or not. This is necessary for backward compatibility to IEEE 802.15.4-2003 and for future use. Even if frame version numbers two and three are reserved, it can be handled by the radio transceiver.

Table 14-23. Frame Control Field – Frame Version Subfield

Frame Control Field Bit Assignments		Description
Frame Version b13 b12	Value	
00	0	Frames are compatible with IEEE 802.15.4-2003
01	1	Frames are compatible with IEEE 802.15.4-2006
10	2	Reserved
11	3	Reserved

Bits [15:14]: the “Source Addressing Mode” subfield, with similar meaning as “Destination Addressing Mode”.

The addressing field description bits of the FCF (Bits 0–2, 3, 6, 10–15) affect the AT86RF212B Frame Filter.

Related Links

14.2.1 Operating Modes
14.2.2.3 RX_AACK_ON-Receive with Automatic ACK
14.2.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios
14.3.2 Frame Filter
14.8.23 XAH_CTRL_1
14.8.38 CSMA_SEED_1

14.3.1.2.3 Frame Compatibility between IEEE 802.15.4 2003 and IEEE 802.15.4 2006

All unsecured frames according to IEEE 802.15.4-2006 are compatible with unsecured frames compliant with IEEE 802.15.4-2003 with two exceptions: a coordinator realignment command frame with the “Channel Page” field present (see IEEE 802.15.4-2006 [2], Section 7.3.8) and any frame with a MAC Payload field larger than *aMaxMACSafePayloadSize octets*.

Compatibility for secured frames is shown in the table below, which identifies the security operating modes for IEEE 802.15.4-2003 and IEEE 802.15.4-2006.

Table 14-24. Frame Control Field – Security and Frame Version

Frame Control Field Bit Assignments		Description
Security Enabled b3	Frame Version b13 b12	
0	00	No security. Frames are compatible between IEEE 802.15.4-2003 and IEEE 802.15.4-2006.
0	01	No security. Frames are not compatible between IEEE 802.15.4-2003 and IEEE 802.15.4-2006.
1	00	Secured frame formatted according to IEEE 802.15.4-2003. This frame type is not supported in IEEE 802.15.4-2006.
1	01	Secured frame formatted according to IEEE 802.15.4-2006.

14.3.1.2.4 Sequence Number

The one-octet sequence number following the FCF identifies a particular frame, so that duplicated frame transmissions can be detected. While operating in RX_AACK mode, the content of this field is copied from the frame to be acknowledged into the acknowledgment frame.

14.3.1.2.5 Addressing Fields

The addressing fields of the MPDU are used by the AT86RF212B for address matching indication. The destination address (if present) is always first, followed by the source address (if present). Each address field consists of the PAN-ID and a device address. If both addresses are present, and the “PAN ID compression” subfield in the FCF is set to one, the source PAN-ID is omitted.

Note that in addition to these general rules, IEEE 802.15.4 further restricts the valid address combinations for the individual possible MAC frame types. For example, the situation where both addresses are omitted (source addressing mode = 0 and destination addressing mode = 0) is only allowed for acknowledgment frames. The address filter in the AT86RF212B has been designed to apply to IEEE 802.15.4 compliant frames. It can be configured to handle other frame formats and exceptions.

14.3.1.2.6 Auxiliary Security Header Field

The Auxiliary Security Header specifies information required for security processing and has a variable length. This field determines how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (see IEEE 802.15.4-2006 [2], Section 7.6.1). This field shall be present only if the Security Enabled subfield b3 is set to one. For details of its structure, see IEEE 802.15.4-2006, Section 7.6.2 Auxiliary security header.

Related Links

14.3.1.2.3 Frame Compatibility between IEEE 802.15.4 2003 and IEEE 802.15.4 2006

14.3.1.2.7 MAC Service Data Unit (MSDU)

This is the actual MAC payload. It is usually structured according to the individual frame type. A description can be found in IEEE 802.15.4-2006, Section 5.5.3.2.

14.3.1.2.8 MAC Footer (MFR) Fields

The MAC footer consists of a two octet Frame Checksum (FCS).

Related Links

14.3.3 Frame Check Sequence (FCS)

14.3.2 Frame Filter

Frame Filtering is a procedure that evaluates whether or not a received frame matches predefined criteria, like source or destination address or frame types. A filtering procedure as described in IEEE 802.15.4-2006 Section 7.5.6.2 (Third level of filtering) is applied to the frame to accept a received frame and to generate the address match interrupt IRQ_5 (AMI).

The AT86RF212B Frame Filter passes only frames that satisfy all of the following requirements/rules (quote from IEEE 802.15.4-2006, Section 7.5.6.2):

1. The Frame Type subfield shall not contain a reserved frame type.
2. The Frame Version subfield shall not contain a reserved value.
3. If a destination PAN identifier is included in the frame, it shall match *macPANId* or shall be the broadcast PAN identifier (0xFFFF).
4. If a short destination address is included in the frame, it shall match either *macShortAddress* or the broadcast address (0xFFFF). Otherwise, if an extended destination address is included in the frame, it shall match *aExtendedAddress*.
5. If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match *macPANId* unless *macPANId* is equal to 0xFFFF, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
6. If only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is the PAN coordinator and the source PAN identifier matches *macPANId*.

In addition the AT86RF212B has two additional requirements:

1. The frame type shall indicate that the frame is not an acknowledgment (ACK) frame.
2. At least one address field must be present.

Address match, indicated by interrupt IRQ_5 (AMI), is further controlled by the content of subfields of the frame control field of a received frame according to the following rule:

If Destination Addressing Mode is 0/1 and Source Addressing Mode is zero, no interrupt IRQ_5 (AMI) is generated. This effectively causes all acknowledgement frames not to be announced, which would otherwise always pass the filter, regardless of whether they are intended for this device or not.

For backward compatibility to IEEE 802.15.4-2003 third level filter rule two (Frame Version) can be disabled by the AACK_FVN_MODE bits in the CSMA_SEED_1 register (CSMA_SEED_1.AACK_FVN_MODE).

Frame filtering is available in Extended and Basic Operating Mode. A frame that passes the Frame Filter generates the interrupt IRQ_5 (AMI) if not masked.

Notes:

1. Filter rule one is affected by the AACK_FLTR_RES_FT and AACK_UPLD_RES_FT bits in the XAH_CTRL_1 register (XAH_CTRL_1.AACK_FLTR_RES_FT and XAH_CTRL_1.AACK_UPLD_RES_FT).
2. Filter rule two is affected by register bits CSMA_SEED_1.AACK_FVN_MODE.

Related Links

14.3.1.2.2 Frame Control Field (FCF)
14.8.38 CSMA_SEED_1
14.8.23 XAH_CTRL_1

14.3.2.1 Configuration

The Frame Filter is configured by setting the appropriate address variables and several additional properties as described in the table below.

Table 14-25. Frame Filter Configuration

Register Address	Register Bits	Register / Bit Name	Description
0x20,0x21 0x22,0x23 0x24 … 0x2B		SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7	Set *macShortAddress*, *macPANId* , and *aExtendedAddress* as described in [2].
0x17	1	XAH_CTRL_1.AACK_PROM_MODE	*0:* Disable promiscuous mode. *1*: Enable promiscuous mode.

.....continued

Register Address	Register Bits	Register / Bit Name	Description
0x17	4	XAH_CTRL_1.AACK_UPLD_RES_FT	Enable reserved frame type reception, needed to receive non-standard compliant frames. *0*: Disable reserved frame type reception. *1*: Enable reserved frame type reception.
0x17	5	XAH_CTRL_1.AACK_FLTR_RES_FT	Filter reserved frame types like data frame type, needed for filtering of non-standard compliant frames. *0*: Disable reserved frame types filtering. *1*: Enable reserved frame types filtering.
0x2E	3	CSMA_SEED_1.AACK_I_AM_COORD	*0*: Device is not PAN coordinator. *1*: Device is PAN coordinator.
0x2E	7:6	CSMA_SEED_1.AACK_FVN_MODE	Controls the ACK behavior, depending on FCF frame version number. *b00*: Acknowledges only frames with version number 0, that is according to IEEE 802.15.4-2003 frames. *b01*: Acknowledges only frames with version number 0 or 1, that is frames according to IEEE 802.15.4-2006. *b10*: Acknowledges only frames with version number 0 or 1 or 2. *b11*: Acknowledges all frames, independent of the FCF frame version number.

Related Links

14.3.2.2 Handling of Reserved Frame Types
14.8.38 CSMA_SEED_1
14.8.35 IEEE_ADDR
14.8.33 PAN_ID_0
14.8.34 PAN_ID_1
14.8.31 SHORT_ADDR_0
14.8.32 SHORT_ADDR_1
14.8.23 XAH_CTRL_1

14.3.2.2 Handling of Reserved Frame Types

Reserved frame types are treated according to the AACK_UPLD_RES_FT and AACK_FLTR_RES_FT bits in the XAH_CTRL_1 register (XAH_CTRL_1.AACK_UPLD_RES_FT and XAH_CTRL_1.AACK_FLTR_RES_FT) with three options:

1. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 0:
   Any non-corrupted frame with a reserved frame type is indicated by an IRQ_3 (TRX_END) interrupt. No further address filtering is applied on those frames. An IRQ_5 (AMI) interrupt is never generated and the acknowledgment subfield is ignored.
2. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 1:
   If AACK_FLT_RES_FT = 1 any frame with a reserved frame type is filtered by the address filter similar to a data frame as described in the standard. This implies the generation of the IRQ_5 (AMI) interrupts upon

address match. An IRQ_3 (TRX_END) interrupt is only generated if the address matched and the frame was not corrupted. An acknowledgment is only send, when the ACK request subfield was set in the received frame and an IRQ_3 (TRX_END) interrupt occurred.

3. AACK_UPLD_RES_FT = 0:
   Any received frame with a reserved frame type is discarded.

Related Links

14.2.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios

14.8.23 XAH_CTRL_1

14.3.3 Frame Check Sequence (FCS)

The Frame Check Sequence (FCS) is characterized by:

- Indication of bit errors, based on a cyclic redundancy check (CRC) of length 16 bit
- A use of International Telecommunication Union (ITU) CRC polynomial
- Automatical evaluation during reception
- Automatical generation during transmission

14.3.3.1 Overview

The FCS is intended for use at the MAC layer to detect corrupted frames at a first level of filtering. It is computed by applying an ITU CRC polynomial to all transferred bytes following the length field (MHR and MSDU fields). The frame check sequence has a length of 16 bit and is located in the last two bytes of a frame.

The AT86RF212B applies an FCS check on each received frame. The FCS check result is stored in the RX_CRC_VALID bit in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID).

On transmission the radio transceiver generates and appends the FCS bytes during the frame transmission. This behavior can be disabled by setting the TX_AUTO_CRC_ON in the TRX_CTRL1 register (TRX_CTRL_1.TX_AUTO_CRC_ON) to '0'.

Related Links

14.3.1.2 MAC Protocol Data Unit (MPDU)

14.8.6 PHY_RSSI

14.8.4 TRX_CTRL_1

14.3.3.2 CRC Calculation

The CRC polynomial used in IEEE 802.15.4 networks is defined by.

$G_{16}(x)=x^{16}+x^{12}+x^{5}+1$

The FCS shall be calculated for transmission using the following algorithm:

$M(x)=b_0{}^{5-1}+b_1{}^{k-2}+...+b_{k-2}+b_{k-1}$

Let be the polynomial representing the sequence of bits for which the checksum is to be computed. Multiply M(x) by , giving the polynomial.

$N(x)=M(x)\cdot x^{16}$

Divide modulo two by the generator polynomial,, to obtain the remainder polynomial,

$R(x)=r_0x^{15}+r_1x^{14}+...+r_{14}+r_{15}$

The FCS field is given by the coefficients of the remainder polynomial, *R(x)*.

> Considering a five octet ACK frame. The MHR field consists of
>
> 0100 0000 0000 0000 0101 0110.
>
> The leftmost bit (b0) is transmitted first in time. The FCS is in this case
>
> 0010 0111 1001 1110.
>
> The leftmost bit (r0) is transmitted first in time.

14.3.3.3 Automatic FCS Generation

The automatic FCS generation is enabled by setting the TX_AUTO_CRC_ON bit in the TRX_CTRL_1 register (TRX_CTRL_1.TX_AUTO_CRC_ON) to '1'. This allows the AT86RF212B to compute the FCS autonomously. For a frame with a frame length specified as *N* ($3 \le N \le 127$), the FCS is calculated on the first *N*-2 octets in the Frame Buffer, and the resulting FCS field is transmitted in place of the last two octets from the Frame Buffer.

In RX_AACK mode, when a received frame needs to be acknowledged, the FCS of the ACK frame is always automatically generated by the AT86RF212B, independent of the TX_AUTO_CRC_ON bit setting.

> A frame transmission of length five with TX_AUTO_CRC_ON set, is started with a Frame Buffer write access of five bytes (the last two bytes can be omitted). The first three bytes are used for FCS generation; the last two bytes are replaced by the internally calculated FCS.

Related Links

14.8.4 TRX_CTRL_1

14.3.3.4 Automatic FCS Check

An automatic FCS check is applied on each received frame with a frame length $N \ge 2$. The RX_CRC_VALID bit in the PHY_RSSI register (PHY_RSSI.RX_CRC_VALID) is set if the FCS of a received frame is valid. The register bit is updated when issuing interrupt IRQ_3 (TRX_END) and remains valid until the next TRX_END interrupt caused by a new frame reception. In addition, bit[7] of byte RX_STATUS is set accordingly.

In Extended Operating Mode, the RX_AACK procedure does not accept a frame if the corresponding FCS is not valid, that is no IRQ_3 (TRX_END) interrupt is issued. When operating in TX_ARET mode, the FCS of a received ACK is automatically checked. If it is not correct, the ACK is not accepted.

Related Links

14.1.3.2 Frame Buffer Access Mode
14.2.2.4 TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry
14.8.6 PHY_RSSI

14.3.4 Received Signal Strength Indicator

The Received Signal Strength Indicator is characterized by:

- Minimum RSSI level is RSSIBASE_VAL
- Dynamic range is 87dB
- Minimum RSSI value is 0
- Maximum RSSI value is 28

14.3.4.1 Overview

The RSSI is a 5-bit value indicating the receive power in the selected channel, in steps of 3.1dB. No attempt is made to distinguish IEEE 802.15.4 signals from others, only the received signal strength is evaluated. The RSSI provides the basis for an ED measurement.

Related Links

14.3.5 Energy Detection

14.3.4.2 Reading RSSI

In Basic Operating Modes, the RSSI value is valid in any receive state and is updated at time intervals according to the table below (see parameter t_{RSSI} in the *Block Initialization and Settling Time* table). The current RSSI value can be accessed by reading the RSSI bits in the PHY_RSSI register (PHY_RSSI.RSSI).

Table 14-26. RSSI Update Interval

PHY Mode	Update Interval [µs]
BPSK-20	32
BPSK-40	24

.....continued	
PHY Mode	**Update Interval [µs]**
O-QPSK	8

It is not recommended reading the RSSI value when using the Extended Operating Modes. Instead, the automatically generated ED value should be used.

Related Links

14.2.1.4.6 State Transition Timing Summary

14.3.5 Energy Detection

14.8.6 PHY_RSSI

14.3.4.3 Data Interpretation

The RSSI value is a 5-bit value in a range of zero to 28, indicating the receiver input power in steps of about 3.1dB.

A RSSI value of zero indicates a receiver RF input power less than or equal to PRF ≤ RSSIBASE_VAL. For a RSSI value in the range of one to 28, the RF input power can be calculated as follows:

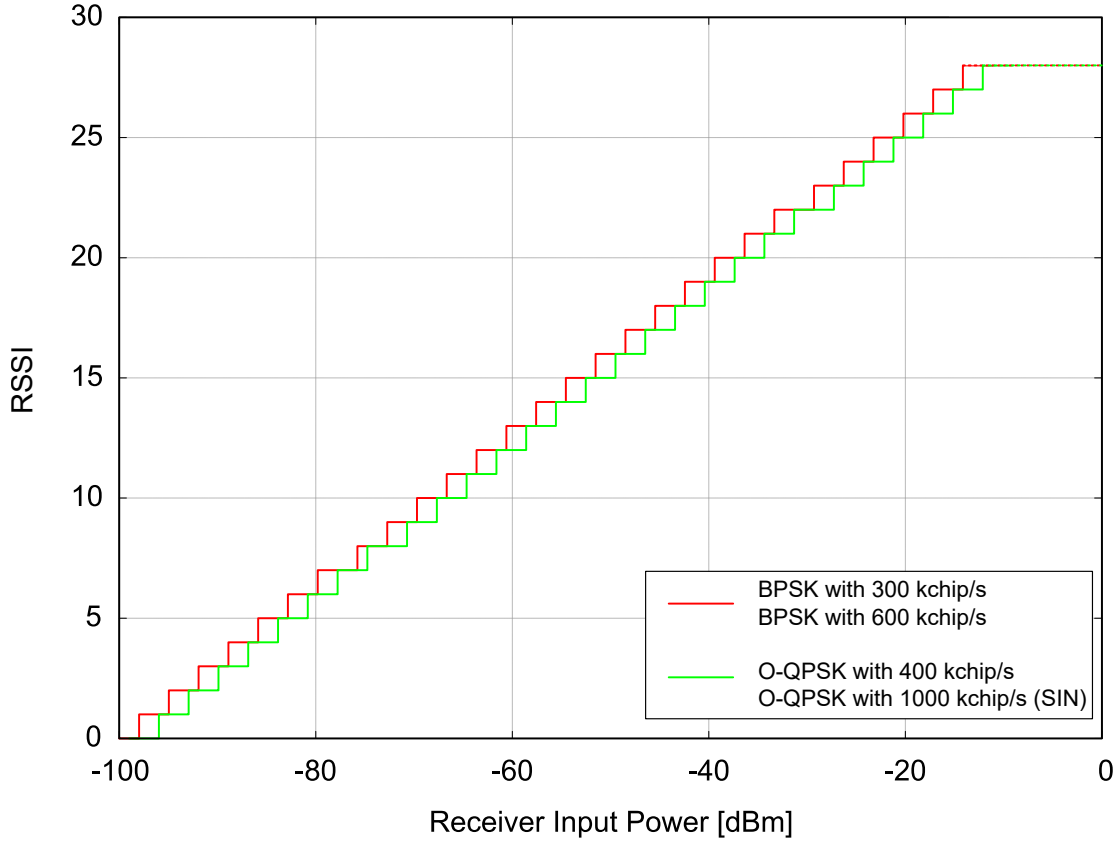
PRF[dBm] = RSSIBASE_VAL[dBm] + 3.1[dB] x RSSI

The value RSSIBASE_VAL itself depends on the PHY mode. For typical conditions, it is shown in the table below.

Table 14-27. RSSI_BASE_VAL

PHY Mode	RSSIBASE_VAL [dBm]
BPSK with 300kchip/s	-100
BPSK with 600kchip/s	-99
O-QPSK with 400kchip/s, SIN and RC-0.2 shaping	-98
O-QPSK with 400kchip/s, RC-0.2 shaping	-98
O-QPSK with 1000kchip/s, SIN shaping	-98
O-QPSK with 1000kchip/s, RC-0.8 shaping	-97

Figure 14-35. Mapping between RSSI Value and Receiver Input Power



Related Links

14.4.1 Physical Layer Modes

14.3.5 Energy Detection

The AT86RF212B Energy Detection (ED) module is characterized by:

- 85 unique energy levels defined
- 1dB resolution
- A measurement time of eight symbol periods for IEEE 802.15.4 compliant data rates

14.3.5.1 Overview

The receiver ED measurement (ED scan procedure) can be used as a part of a channel selection algorithm. It is an estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel. The ED value is calculated by averaging RSSI values over eight symbol periods, with the exception of the High Data Rate Modes.

Related Links

14.4.1.4 Proprietary High Data Rate Modes

14.3.5.2 Measurement Description

There are two ways to initiate an ED measurement,

- Manually by writing an arbitrary value to the PHY_ED_LEVEL register, or
- Automatically after detection of a valid SHR of an incoming frame.

Manually:

For manually initiated ED measurements, the radio transceiver needs to be either in the state RX_ON or BUSY_RX. The end of the ED measurement time (eight symbol periods plus a processing time) is indicated by the interrupt

IRQ_4 (CCA_ED_DONE) and the measurement result is stored in register PHY_ED_LEVEL register, refer to t_{ED} in the *Block Initialization and Settling Time* table.

In order to avoid interference with an automatically initiated ED measurement, the SHR detection can be disabled by setting the RX_PDT_DIS bit in the RX_SYN register (RX_SYN.RX_PDT_DIS).

Note: It is not recommended to manually initiate an ED measurement when using the Extended Operating Mode.

Automatically:

An automated ED measurement is started upon SHR detection. The end of the automated measurement is not signaled by an interrupt.

When using the Basic Operating Mode and standard compliant data rates, a valid ED value (PHY_ED_LEVEL register) of the currently received frame is accessible not later than eight symbol periods after IRQ_2 (RX_START) plus a processing time of 12µs. For High Data Rate Modes, the measurement duration is reduced to two symbol periods plus a processing time of 12µs. The ED value remains valid until a new RX_START interrupt is generated by the next incoming frame or until another ED measurement is initiated.

When using the Extended Operating Mode, it is useful to mask IRQ_2 (RX_START), thus the interrupt cannot be used as timing reference. A successful frame reception is signalized by interrupt IRQ_3 (TRX_END). In this case, the ED value needs to be read within the time span of a next SHR detection plus the ED measurement time in order to avoid overwrite of the current ED value.

Note: The ED result is not updated during the rest of the frame reception, even by requesting an ED measurement manually.

Related Links

14.4.1.4 Proprietary High Data Rate Modes
14.4.2 Receiver (RX)
14.2.1.4.6 State Transition Timing Summary
14.8.7 PHY_ED_LEVEL
14.8.21 RX_SYN

14.3.5.3 Data Interpretation

The PHY_ED_LEVEL is an 8-bit register. The ED_LEVEL value of the AT86RF212B has a valid range from 0x00 to 0x54 with a resolution of 1.03dB. Values 0x55 to 0xFE do not occur and a value of 0xFF indicates the reset value.

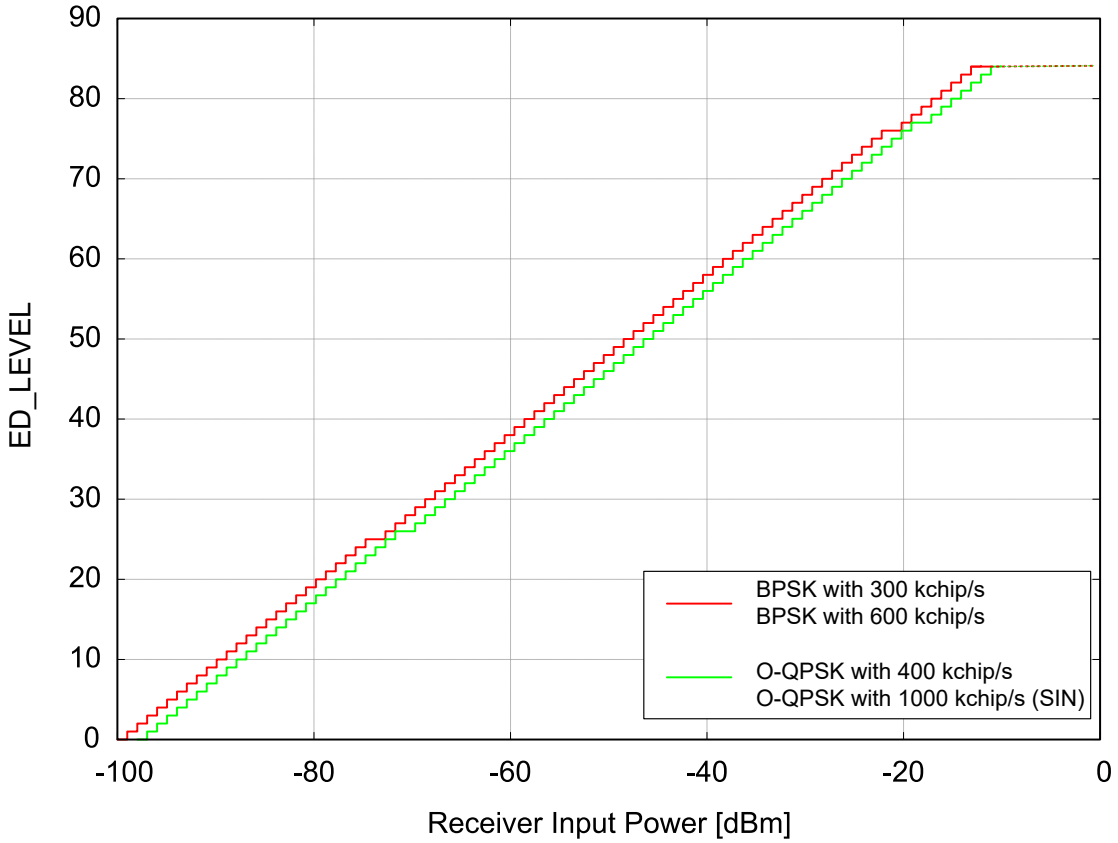
Due to environmental conditions (temperature, voltage, semiconductor parameters, etc.) the calculated ED_LEVEL value has a maximum tolerance of ±6dB, this is to be considered as constant offset over the measurement range.

An ED_LEVEL value of zero indicates a receiver RF input power less than or equal to RSSIBASE_VAL; a value of 84 indicates an input power equal to or larger than RSSIBASE_VAL + 87dB.

The receiver input power can be calculated as follows:

PRF[dBm] = RSSIBASE_VAL[dBm] +1.03[dB] x ED_LEVEL

Figure 14-36. Mapping between ED Value and Receiver Input Power



Related Links

14.3.4.3 Data Interpretation
14.8.7 PHY_ED_LEVEL

14.3.5.4 Interrupt Handling

Interrupt IRQ_4 (CCA_ED_DONE) is issued at the end of a manually initiated ED measurement.

Note: An ED measurement should only be initiated in RX states but not in RX_AACK states. Otherwise, the radio transceiver generates an IRQ_4 (CCA_ED_DONE) without actually performing an ED measurement.

14.3.6 Clear Channel Assessment

The main features of the Clear Channel Assessment (CCA) module are:

- All four modes are available as defined by IEEE 802.15.4-2006 in Section 6.9.9
- Adjustable threshold for energy detection algorithm

14.3.6.1 Overview

A CCA measurement is used to detect a clear channel. Four CCA modes are specified by IEEE 802.15.4-2006:

Table 14-28. CCA Mode Overview

CCA Mode	Description
1	*Energy above threshold*. CCA shall report a busy medium upon detecting any energy above the ED threshold.

.....continued	
CCA Mode	**Description**
2	*Carrier sense only*. CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of an IEEE 802.15.4 compliant signal. The signal strength may be above or below the ED threshold.
0, 3	*Carrier sense with energy above threshold*. CCA shall report a busy medium using a logical combination of • Detection of a signal with the modulation and spreading characteristics of this standard and • Energy above the ED threshold. Where the logical operator may be configured as either OR (mode 0) or AND (mode 3).

14.3.6.2 Configuration and Request

The CCA modes are configurable via the PHY_CC_CCA register.

When in Basic Operating Mode, a CCA request can be initiated manually by setting the CCA_REQUEST bit in the PHY_CC_CCA register (PHY_CC_CCA.CCA_REQUEST) to '1', if the AT86RF212B is in any RX state. The current channel status (CCA_STATUS) and the CCA completion status (CCA_DONE) are accessible through the TRX_STATUS register.

The end of a manually initiated CCA (eight symbol periods plus 12µs processing delay) is indicated by the interrupt IRQ_4 (CCA_ED_DONE).

The CCA_ED_THRES bits in the CCA_THRES register (CCA_THRES.CCA_ED_THRES) defines the receive power threshold of the “energy above threshold” algorithm. The threshold is calculated by:

$P_{CCA\_ED\_THRES}$[dBm] = $RSSI_{BASE\_VAL}$[dBm] + 2.07[dB] x CCA_ED_THRES.

Any received power above this level is interpreted as a busy channel.

Note: It is not recommended to manually initiate a CCA measurement when using the Extended Operating Mode.

Related Links

14.8.9 CCA_THRES
14.8.8 PHY_CC_CCA
14.8.1 TRX_STATUS

14.3.6.3 Data Interpretation

The AT86RF212B current channel status (CCA_STATUS) and the CCA completion status (CCA_DONE) are accessible through register the TRX_STATUS register.

Note: The register bits CCA_DONE and CCA_STATUS are cleared in response to a CCA_REQUEST.

The completion of a measurement cycle is indicated by CCA_DONE = 1. If the radio transceiver detects no signal (idle channel) during the CCA evaluation period, the CCA_STATUS bit is set to one; otherwise, it is set to zero.

When using the “energy above threshold” algorithm, a received power above PCCA_ED_THRES is interpreted as a busy channel.

When using the “carrier sense” algorithm (that is CCA_MODE = 0, 2, and 3), the AT86RF212B reports a busy channel upon detection of a PHY mode specific IEEE 802.15.4 signal above $RSSI_{BASE\_VAL}$. The AT86RF212B is also capable of detecting signals below this value, but the detection probability decreases with decreasing signal power. It is almost zero at the radio transceivers sensitivity level (see parameter P_{SENS} the *Receiver Characteristics*).

Related Links

14.3.4.3 Data Interpretation
15.15.4 Receiver Characteristics
14.8.1 TRX_STATUS

14.3.6.4 Interrupt Handling

Interrupt IRQ_4 (CCA_ED_DONE) is issued at the end of a manually initiated CCA measurement.

Note: A CCA request should only be initiated in Basic Operating Mode receive states. Otherwise the radio transceiver generates an IRQ_4 (CCA_ED_DONE) and sets the register bit CCA_DONE = 1, even though no CCA measurement was performed.

14.3.6.5 Measurement Time

The response time of a manually initiated CCA measurement depends on the receiver state.

In RX_ON state, the CCA measurement is done over eight symbol periods and the result is accessible upon the event IRQ_4 (CCA_ED_DONE) or upon CCA_DONE = 1 in the TRX_STATUS register.

In BUSY_RX state, the CCA measurement duration depends on the CCA mode and the CCA request relative to the detection of the SHR. The end of the CCA measurement is indicated by IRQ_4 (CCA_ED_DONE). The variation of a CCA measurement period in BUSY_RX state is described in the table below.

It is recommended to perform CCA measurements in RX_ON state only. To avoid switching accidentally to BUSY_RX state, the SHR detection can be disabled by setting the RX_PDT_DIS bit in the RX_SYN register (RX_SYN.RX_PDT_DIS). The receiver remains in RX_ON state to perform a CCA measurement until the register bit RX_PDT_DIS is set back to continue the frame reception. In this case, the CCA measurement duration is eight symbol periods.

Table 14-29. CCA Measurement Period and Access in BUSY_RX State

CCA Mode	Request within ED measurement(1)	Request after ED measurement
1	*Energy above threshold*.	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.
2	*Carrier sense only*.	
	CCA result is immediately available after request.	
3	*Carrier sense with Energy above threshold (AND)*.	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.
0	*Carrier sense with Energy above threshold (OR)*.	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.

1. After detecting the SHR, an automated ED measurement is started with a length of eight symbol periods (two symbol periods for high rate PHY modes). This automated ED measurement must be finished to provide a result for the CCA measurement. Only one automated ED measurement per frame is performed.

Related Links

14.4.1.4 Proprietary High Data Rate Modes
14.4.2 Receiver (RX)
14.8.21 RX_SYN
14.8.1 TRX_STATUS

14.3.7 Listen Before Talk (LBT)

14.3.7.1 Overview

Equipment using the AT86RF212B shall conform to the established regulations. With respect to the regulations in Europe, CSMA-CA based transmission according to IEEE 802.15.4 is not appropriate. In principle, transmission is subject to low duty cycles (0.1% to 1%). However, according to [6], equipment employing listen before talk (LBT) and adaptive frequency agility (AFA) does not have to comply with duty cycle conditions. Hence, LBT can be attractive in order to reduce network latency.

Minimum Listening Time

A device with LBT needs to comply with a minimum listening time, refer to Section 9.1.1.2 of [6]. Prior transmission, the device must listen for a receive signal at or above the LBT threshold level to determine whether the intended channel is available for use, unless transmission is pursuing acknowledgement.

A device using LBT needs to listen for a fixed period of at least 5ms. If the channel is free after this period, transmission may immediately commence (that is no CSMA is required). Otherwise, a new minimum listening period of a randomly selected time span between 5ms and 10ms is required. The time resolution shall be approximately 0.5ms. The last step needs to be repeated until a free channel is available.

LBT Threshold

According to [6], the maximum LBT threshold for an IEEE 802.15.4 signal is presumably -82dBm, assuming a channel spacing of 1MHz.

14.3.7.2 LBT Mode

The AT86RF212B supports the previously described LBT specific listening mode when operating in the Extended Operating Mode.

In particular, during Transmit with Automatic Retransmission (TX_ARET), the CSMA-CA algorithm can be replaced by the LBT listening mode when setting the CSMA_LBT_MODE bit in the XAH_CTRL_1 register (XAH_CTRL_1.CSMA_LBT_MODE). In this case, however, the MAX_CSMA_RETRIES bits in the XAH_CTRL_0 register (XAH_CTRL_0.MAX_CSMA_RETRIES) as well as the MIN_BE and MAX_BE bits in the CSMA_BE register (CSMA_BE.MIN_BE and CSMA_BE.MAX_BE) are ignored, implying that the listening mode will sustain unless a clear channel has been found or the TX_ARET transaction will be canceled. The latter can be achieved by setting the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD) to either FORCE_PLL_ON or FORCE_TRX_OFF, the value of the TRAC_STATUS bits in the TRX_STATE register (TRAC_STATUS.TRX_STATE) is not meaningful in this case. All other aspects of TX_ARET remain unchanged.

The LBT threshold can be configured in the same way as for CCA, that is via the CCA_MODE bits in the PHY_CC_CCA register (PHY_CC_CCA.CCA_MODE) and the CCA_ED_THRES bits in the CCA_THRES register (CCA_THRES.CCA_ED_THRES), refer to Section 8.6.

Related Links

14.2.2 Extended Operationg Mode
14.2.2.4 TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry
14.3.6 Clear Channel Assessment
14.8.9 CCA_THRES
14.8.39 CSMA_BE
14.8.8 PHY_CC_CCA
14.8.2 TRX_STATE
14.8.36 XAH_CTRL_0
14.8.23 XAH_CTRL_1

14.3.8 Link Quality Indication (LQI)

The IEEE 802.15.4 standard defines the LQI as a characterization of the strength and/or quality of a received frame. The use of the LQI result by the network or application layer is not specified in this standard. The LQI value shall be an integer ranging from zero to 255, with at least eight unique values. The minimum and maximum LQI values (0x00 and 0xFF) should be associated with the lowest and highest quality compliant signals, respectively, and LQI values in between should be uniformly distributed between these two limits.

14.3.8.1 Overview

During symbol detection within frame reception, the AT86RF212B uses correlation results of multiple symbols in order to compute an estimate of the LQI value. This is motivated by the fact that the mean value of the correlation result is inversely related to the probability of a detection error.

LQI computation is automatically performed for each received frame, once the SHR has been detected. LQI values are integers ranging from zero to 255 as required by the IEEE 802.15.4 standard.

14.3.8.2 Obtaining the LQI Value

The LQI value is available, once the corresponding frame has been completely received. This is indicated by the interrupt IRQ_3 (TRX_END). The value can be obtained by means of a frame buffer read access.

Related Links

14.1.3.2 Frame Buffer Access Mode

14.3.8.3 Data Interpretation

The reason for a low LQI value can be twofold: a low signal strength and/or high signal distortions, for example by interference and/or multipath propagation. High LQI values, however, indicate a sufficient signal strength and low signal distortions.

Notes:

1. The LQI value is almost always 255 for scenarios with very low signal distortions and a signal strength much greater than the sensitivity level. In this case, the packet error rate tends towards zero and increase of the signal strength, that is by increasing the transmission power, cannot decrease the error rate any further. Received signal strength indication (RSSI) or energy detection (ED) can be used to evaluate the signal strength and the link margin.
2. The received signal power as indicated by received signal strength indication (RSSI) value or energy detection (ED) value of the AT86RF212B do not characterize the signal quality and the ability to decode a signal.

ZigBee networks often require identification of the “best” routing between two nodes. LQI and RSSI/ED can be applied, depending on the optimization criteria. If a low frame error rate (corresponding to a high throughput) is the optimization criteria, then the LQI value should be taken into consideration. If, however, the target is a low transmission power, then the RSSI/ED value is also helpful.

Various combinations of LQI and RSSI/ED are possible for routing decisions. As a rule of thumb, information on RSSI/ED is useful in order to differentiate between links with high LQI values. However, transmission links with low LQI values should be discarded for routing decisions, even if the RSSI/ED values are high, since it is merely an information about the received signal strength, whereas the source can be an interferer.

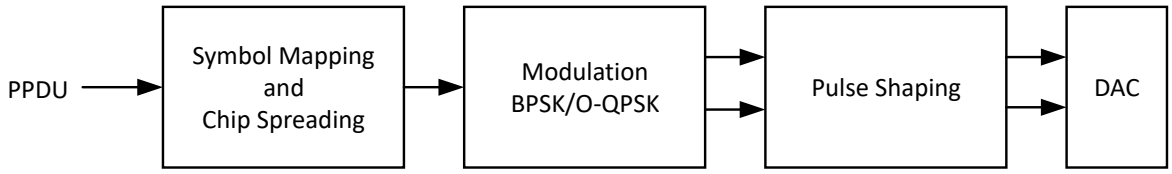
14.4 Module Description

14.4.1 Physical Layer Modes

14.4.1.1 Spreading, Modulation and Pulse Shaping

The AT86RF212B supports various physical layer (PHY) modes independent of the RF channel selection. Symbol mapping along with chip spreading, modulation and pulse shaping are a part of the digital base band processor.

Figure 14-37. Base Band Transmitter Architecture



The combination of spreading, modulation and pulse shaping are restricted to several combinations, as shown in the table below.

The AT86RF212B is fully compliant to the IEEE 802.15.4 low data rate modes of 20 kb/s or 40 kb/s, employing binary phase-shift keying (BPSK) and spreading with a fixed chip rate of 300 kchip/s or 600 kchip/s, respectively. The symbol rate is 20 ksymbol/s or 40 ksymbol/s, respectively. In both cases, pulse shaping is approximating a raised cosine filter with roll-off factor 1.0 (RC-1.0).

For optional data rates according to IEEE 802.15.4-2006, offset quadrature phase-shift keying (O-QPSK) is supported by the AT86RF212B with a fixed chip rate of either 400 kchip/s or 1000 kchip/s.

At a chip rate of 400 kchip/s, there is a choice between two different Pulse Shaping modes. One Pulse Shaping mode uses a combination of both half-sine shaping (SIN) and raised cosine filtering with roll-off factor 0.2 (RC-0.2)

according to IEEE 802.15.4-2006 [2] for the 868.3 MHz band. The other uses raised cosine filtering with roll-off factor 0.2 (RC-0.2).

At a chip rate of 1000 kchip/s, pulse shaping is either half-sine filtering (SIN), as specified in IEEE 802.15.4-2006 [2], or, alternatively, raised cosine filtering with roll-off factor 0.8 (RC-0.8) as specified in IEEE 802.15.4c™-2009 [3] and IEEE 802.15.4-2011 [4].

For O-QPSK, the AT86RF212B supports spreading according to IEEE 802.15.4-2006 with data rates of either 100 kb/s or 250 kb/s depending on the chip rate, leading to a symbol rate of either 25 ksymbol/s or 62.5 ksymbol/s, respectively.

Additionally, the AT86RF212B supports two more spreading codes for O-QPSK with shortened code lengths. This leads to higher but non-IEEE 802.15.4 compliant data rates for PSDU transmission at 200, 400, 500 and 1000 kb/s.

Table 14-30. Modulation and Pulse Shaping

Modulation	Chip Rate [kchip/s]	Supported Data Rate for PPDU Header [kb/s]	Supported Data Rates for PSDU [kb/s]	Pulse Shaping
BPSK	300	20	20	RC-1.0
	600	40	40(1)	RC-1.0
O-QPSK	400	100	100, 200, 400	SIN and RC-0.2
	400	100	100, 200, 400	RC-0.2
	1000	250	250, 500(1), 1000	SIN
	1000	250	250, 500(1), 1000	RC-0.8

1. Support of two different spreading codes.

Related Links

14.4.1.4 Proprietary High Data Rate Modes

14.4.1.2 Configuration

The PHY mode can be selected by setting the appropriate BPSK_OQPSK, SUB_MODE, OQPSK_DATA_RATE, and ALT_SPECTRUM bits in the TRX_CTRL_2. During configuration, the transceiver needs to be in TRX_OFF state.

Related Links

14.8.12 TRX_CTRL_2

14.4.1.3 Symbol Period

Within IEEE 802.15.4 and, accordingly, within this document, time references are often specified in units of symbol periods, leading to a PHY mode independent description. The table below shows the duration of the symbol period.

Table 14-31. Duration of the Symbol Period

Modulation	PSDU Data Rate [kb/s]	Duration of Symbol Period [µs]
BPSK	20	50
	40	25
O-QPSK	100, 200, 400	40
	250, 500, 1000	16

1. For the proprietary High Data Rate Modes, the symbol period is (by definition) the same as the symbol period of the corresponding base mode.

Related Links

14.4.1.4 Proprietary High Data Rate Modes

14.4.1.4 Proprietary High Data Rate Modes

The main features are:

- High data rates up to 1000kb/s
- Support of Basic and Extended Operating Mode
- Reduced ACK timing (optional)

Related Links

14.2.1 Operating Modes
14.2.2 Extended Operationg Mode

14.4.1.4.1 Overview

The AT86RF212B supports alternative data rates of 200, 400, 500, and 1000kb/s for applications not necessarily targeting IEEE 802.15.4 compliant networks.

The High Data Rate Modes utilize the same RF channel bandwidth as the IEEE 802.15.4-2006 sub-1GHz O-QPSK modes. Higher data rates are achieved by using the modified O-QPSK spreading codes having reduced code lengths. The lengths are reduced by the factor of two or by the factor of four.

For O-QPSK with 400kchip/s, this leads to a data rate of 200kb/s (2-fold) and 400kb/s (4-fold), respectively.

For O-QPSK with 1000kchip/s, the resulting data rate is 500kb/s (2-fold) and 1000kb/s (4-fold), respectively.

Due to the decreased spreading factor, the sensitivity of the receiver is reduced. The P_{SENS} parameter in the *Receiver Characteristics* shows typical values of the sensitivity for different data rates.

Related Links

15.15.4 Receiver Characteristics

14.4.1.4.2 High Data Rate Frame Structure

In order to allow robust frame synchronization, the AT86RF212B high data rate modulation is restricted to the PSDU part only. The PPDU header (the preamble, the SFD, and the PHR field) are transmitted with a rate of either 100kb/s or 250kb/s (basic rates).

Figure 14-38. High Date Rate Frame Structure

Due to the overhead caused by the PPDU header and the FCS, the effective data rate is less than the selected data rate, depending on the length of the PSDU. A graphical representation of the effective data rate is shown in the figure below.

Figure 14-39. Effective Data Rate “B” for O-QPSK High Data Rate Modes

Consequently, high data rate transmission is useful for large PSDU lengths due to the higher effective data rate, or in order to reduce the power consumption of the system.

14.4.1.4.3 High Data Rate Mode Options

Reduced Acknowledgment Time

If the AACK_ACK_TIME bit in the XAM_CTRL_1 register (XAH_CTRL_1.AACK_ACK_TIME) is set, the acknowledgment time is reduced to the duration of two symbol periods for 200 and 400kb/s data rates, and to three symbol periods for 500 and 1000kb/s data rates. The reduced acknowledgment time is untouched in IEEE 802.15.4. Otherwise, it defaults to 12 symbol periods according to IEEE 802.15.4.

Receiver Sensitivity Control

The different data rates between PPDU header (SHR and PHR) and PHY payload (PSDU) cause a different sensitivity between header and payload. This can be adjusted by defining sensitivity threshold levels of the receiver. With a sensitivity threshold level set, the AT86RF212B does not synchronize to frames with an RSSI level below that threshold. The sensitivity threshold is configured by the RX_PDT_LEVEL bits in the RX_SYN register (RX_SYN.RX_PDT_LEVEL).

Scrambler

For data rates 400kb/s and 1000kb/s, additional chip scrambling is applied per default in order to mitigate data dependent spectral properties. Scrambling can be disabled if the OQPSK_SCRAM_EN bit in the TRX_CTRL_2 register (TRX_CTRL_2.OQPSK_SCRAM_EN) is set to zero.

Energy Detection

The Energy Detection (ED) measurement time span is eight symbol periods according to IEEE 802.15.4. For frames operated at a higher data rate, the automated measurement duration is reduced to two symbol periods taking reduced frame durations into account. This means, the ED measurement time is 80µs for modes 200kb/s and 400kb/s, and 32µs for modes 500kb/s and 1000kb/s. For manually initiated ED measurements in these modes, the measurement time is still eight symbol periods.

Carrier Sense

For clear channel assessment, IEEE 802.15.4-2006 specifies several modes which may either apply “energy above threshold” or “carrier sense” (CS) or a combination of both. Since signals of the High Data Rate Modes are not compliant to IEEE 802.15.4-2006, CS is not supported when the AT86RF212B is operating in these modes. However, “energy above threshold” is supported.

Link Quality Indicator (LQI)

For the High Data Rate Modes, the link quality value does not contain useful information and should be discarded.

Related Links

14.3.5.2 Measurement Description
14.8.21 RX_SYN
14.8.12 TRX_CTRL_2
14.8.23 XAH_CTRL_1

14.4.2 Receiver (RX)

14.4.2.1 Overview

The AT86RF212B transceiver is split into an analog radio front-end and a digital domain. Referring to the receiver part of the analog domain, the differential RF signal is amplified by a low noise amplifier (LNA) and split into quadrature signals by a poly-phase filter (PPF). Two mixer circuits convert the quadrature signal down to an intermediate frequency. Channel selectivity is achieved by an integrated band-pass filter (BPF). The subsequent analog-to-digital converter (ADC) samples the receive signal and additionally generates a digital RSSI signal. The ADC output is then further processed by the digital baseband receiver (RX BBP), which is part of the digital domain.

The BBP performs further filtering and signal processing. In RX_ON state, the receiver searches for the synchronization header. Once the synchronization is established and the SFD is found, the received signal is demodulated and provided to the Frame Buffer. Upon synchronization the receiver performs a state change from RX_ON to BUSY_RX which is indicated by the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS). Once the frame is received, the receiver switches back to RX_ON in the listen mode on the selected channel. A similar scheme applies to the Extended Operating Mode.

The receiver is designed to handle reference oscillator accuracies up to ±60ppm; refer to the f_{SRD} parameter in the *General RF Specifications* section. This results in the estimation and correction of frequency and symbol rate errors up to ±120ppm.

Several status information are generated during the receive process: LQI, ED, and RX_STATUS. They are automatically appended during Frame Read Access. Some information is also available through register access, for example the PHY_ED_LEVEL.ED_LEVEL and FCS correctness with the PHY_RSSI.RX_CRC_VALID.

The Extended Operating Mode of the AT86RF212B supports frame filtering and pending data indication.

Related Links

4.3 Transceiver Circuit Description
14.1.4 Radio Transceiver Status Information
14.1.3.2 Frame Buffer Access Mode
14.8.7 PHY_ED_LEVEL
14.8.6 PHY_RSSI
14.8.1 TRX_STATUS

14.4.2.2 Frame Receive Procedure

The frame receive procedure, including the radio transceiver setup for reception and reading PSDU data from the Frame Buffer, is described in the *Frame Receive Procedure* section.

Related Links

14.5.1 Frame Receive Procedure

14.4.2.3 Configuration

In Basic Operating Mode, the receiver is enabled by writing command RX_ON to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD) in states TRX_OFF or PLL_ON. In Extended Operating Mode, the receiver is enabled for RX_AACK operation from state PLL_ON by writing the command RX_AACK_ON.

There is no additional configuration required to receive IEEE 802.15.4 compliant frames in Basic Operating Mode. However, the frame reception in the AT86RF212B Extended Operating Mode requires further register configurations.

For specific applications, the receiver can additionaly be configured to handle critical environment to simplify the interaction with the microcontroller, or to operate in different data rates.

There are scenarios where CSMA-CA is not used before a transmission or where CSMA-CA is not really reliable, for example in hidden node scenarios. As two transceivers compete for the use of one channel they may interfere with each other which may produce unreliable transmission. Receiver Override can be used to cope with such scenarios. The level of interference (which can be caused by a new incoming frame) is continuously measured while decoding a frame. The synchronization to the potential new frame starts if the interference level does not allow for a reliable detection.

The AT86RF212B receiver has an outstanding sensitivity performance. At certain environmental conditions or for High Data Rate Modes it may be useful to manually decrease this sensitivity. This is achieved by adjusting the synchronization header detector threshold using register the RX_PDT_LEVEL bits in the RX_SYN register (RX_SYN.RX_PDT_LEVEL). Received signals with a RSSI value below the threshold do not activate the demodulation process.

Furthermore, at times it may be useful to protect a received frame against overwriting by a new subsequent data frame, when the receive data buffer has not been read on time. A Dynamic Frame Buffer Protection is enabled with the RX_SAFE_MODE bit in the TRX_CTRL_2 register (TRX_CTRL_2.RX_SAFE_MODE) set. The receiver remains in RX_ON or RX_AACK_ON state until the whole frame is uploaded by the microcontroller, indicated by pin 23 (/SEL) = H during the SPI Frame Receive Mode. The Frame Buffer content is only protected if the FCS is valid.

A Static Frame Buffer Protection is enabled with the RX_PDT_DIS bit in the RX_SYN register (RX_SYN.RX_PDT_DIS) set. The receiver remains in RX_ON or RX_AACK_ON state and no further SHR is detected until the register bit RX_PDT_DIS is set back.

Related Links

14.4.1.4 Proprietary High Data Rate Modes
14.2.1 Operating Modes
14.2.2 Extended Operationg Mode
14.2.2.2 Configuration
14.8.21 RX_SYN
14.8.12 TRX_CTRL_2
14.8.2 TRX_STATE

14.4.3 Transmitter (TX)

14.4.3.1 Overview

The AT86RF212B transmitter utilizes a direct up-conversion topology. The digital transmitter (TX BBP) generates the in-phase (I) and quadrature (Q) component of the modulation signal. A digital-to-analog converter (DAC) forms the analog modulation signal. A quadrature mixer pair converts the analog modulation signal to the RF domain. The power amplifier (PA) provides signal power delivered to the differential antenna pins (RFP, RFN). Both, the LNA of the receiver input and the PA of the transmitter output are internally connected to the bidirectional differential antenna pins so that no external antenna switch is needed.

Using the default settings, the PA incorporates an equalizer to improve its linearity. The enhanced linearity keeps the spectral side lobes of the transmit spectrum low in order to meet the requirements of the European 868.3MHz band.

If the PA boost mode is turned on, the equalizer is disabled. This allows to deliver a higher transmit power of up to +11dBm at the cost of higher spectral side lobes and higher harmonic power.

In Basic Operating Mode, a transmission is started from PLL_ON state by either writing TX_START to the TRX_CMD bits in the TRX_STATE regoster (TRX_STATE.TRX_CMD) or by a rising edge of pin 11 (SLP_TR).

In Extended Operating Modes, a transmission might be started automatically depending on the transaction phase of either RX_AACK or TX_ARET.

Related Links

14.2.1 Operating Modes
14.2.2 Extended Operationg Mode
14.8.2 TRX_STATE

14.4.3.2 Frame Transmit Procedure

The frame transmit procedure, including writing PSDU data into the Frame Buffer and initiating a transmission, is described in the *Radio Transceiver Usage - Frame Transmit Procedure* section.

Related Links

14.5.2 Frame Transmit Procedure

14.4.3.3 Spectrum Mask

The AT86RF212B can be operated in different frequency bands, using different power levels, modulation schemes, chip rates, and pulse shaping filters. The occupied bandwidth of the transmit signal depends on the chosen mode of operation. Values listed in table below are based on a default power setting of +5dBm and usage of the Continuous Transmission Test Mode with Frame Buffer content {0x01, 0x00}.

Knowledge of modulation bandwidth, power spectrum, and side lobes is essential for proper system setup that produces non-overlapping channel spacing.

Table 14-32. Physical Layer Mode and Occupied Bandwidth

PHY Mode	99% Occupied Bandwidth [kHz]	6dB Bandwidth [kHz]	20dB Bandwidth [kHz]
Reference	ETSI EN 300 220 [6]	FCC 15.247 [5]	FCC 15.247 [5]
Detector	RMS	Peak/MaxHold	Peak/MaxHold
Span	2 MHz	2 MHz	2 MHz
RBW	100 kHz	5 % of bandwidth	1% of bandwidth
VBW	1 MHz	3 x RBW	3 x RBW
Sweep	500 ms	AUTO	AUTO
BPSK-20	445	295	430
BPSK-40	775	570	850
BPSK-40-ALT	805	620	815

.....continued			
PHY Mode	**99% Occupied Bandwidth [kHz]**	**6dB Bandwidth [kHz]**	**20dB Bandwidth [kHz]**
OQPSK-SIN-RC-100	450	260	340
OQPSK -RC-100	490	355	365
OQPSK-SIN-250	1190	645	1210
OQPSK-RC-250	1245	850	1220

The following figures show the power spectra for the different modes listed in the table above. The spectra were captured using default settings of AT86RF212B. The resolution bandwidth of the spectrum analyzer was set to 30kHz; the video bandwidth was set to 10kHz. For the OQPSK-SIN-250 modulation and OQPSK-RC-250 modulation the resolution bandwidth of the spectrum analyzer was set to 100kHz; the video bandwidth was set to 30kHz.

Figure 14-40. Spectrum of BPSK-20

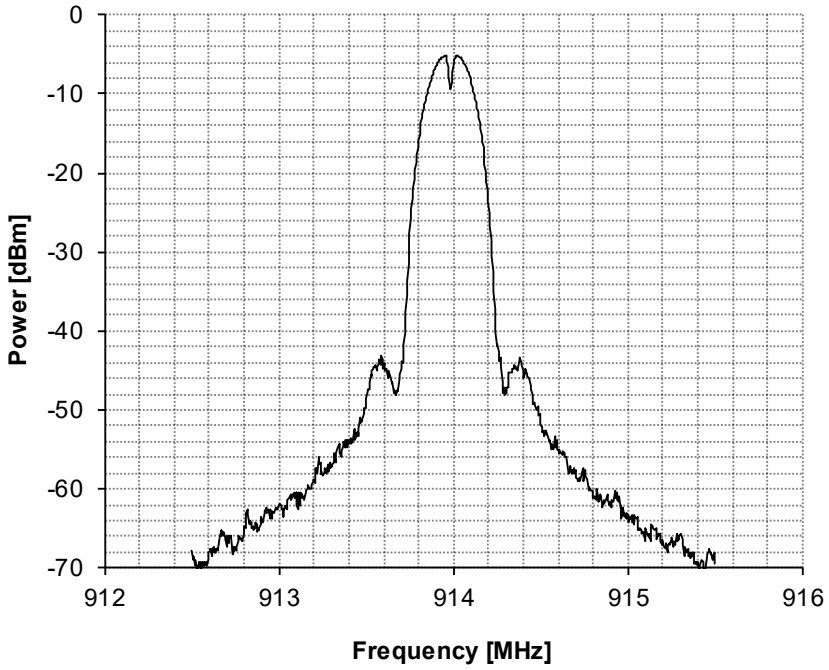


Figure 14-41. Spectrum of BPSK-40

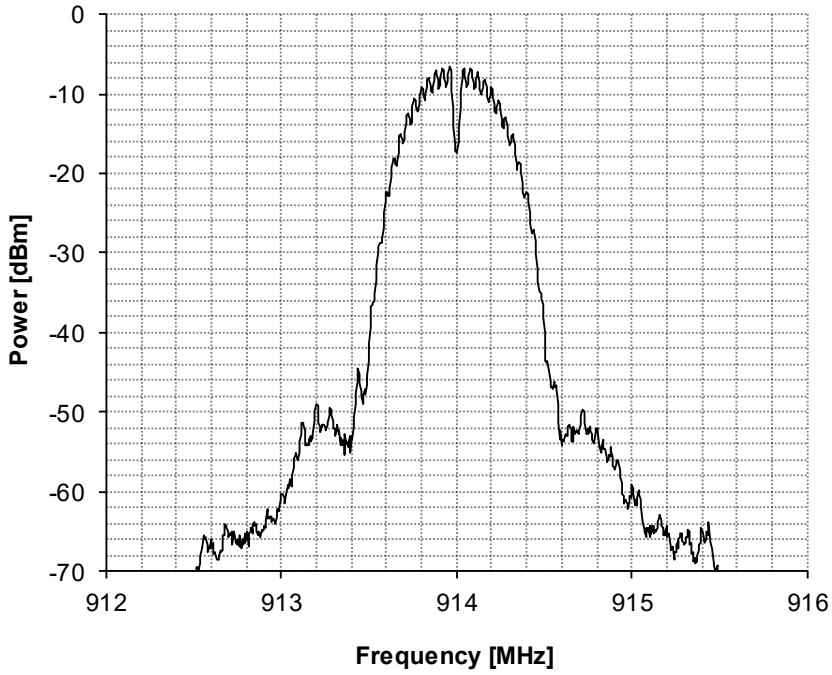


Figure 14-42. Spectrum of BPSK-40-ALT

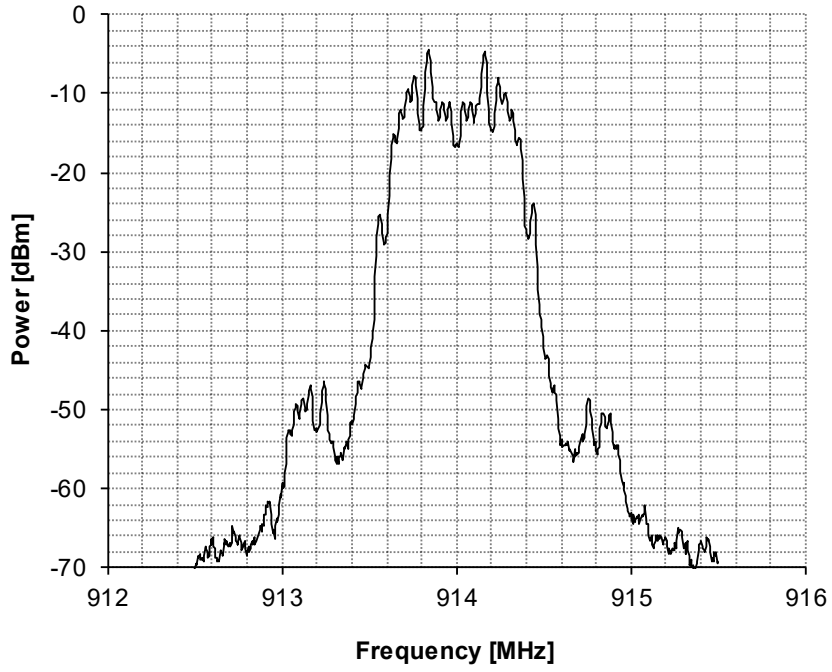


Figure 14-43. Spectrum of OQPSK-SIN-RC-100

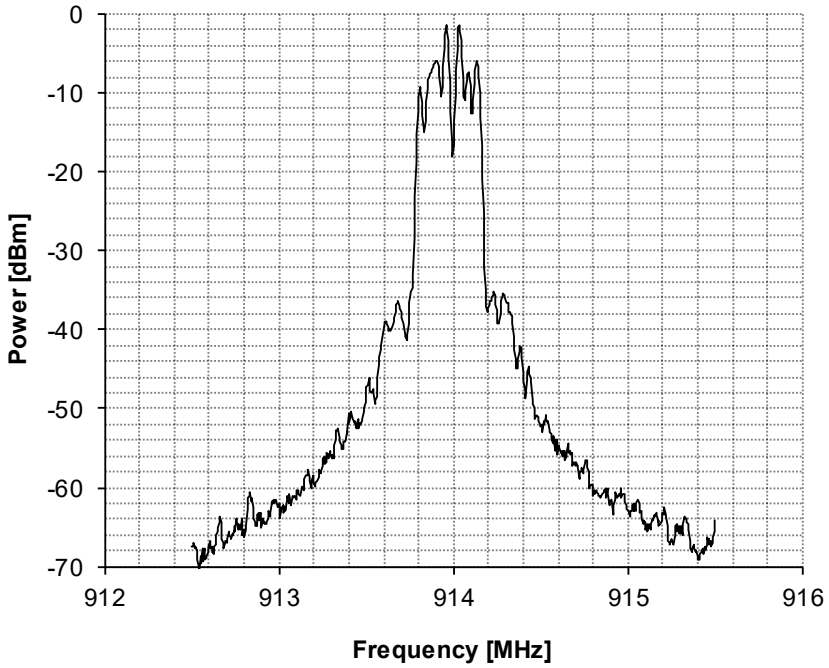


Figure 14-44. Spectrum of OQPSK-RC-100

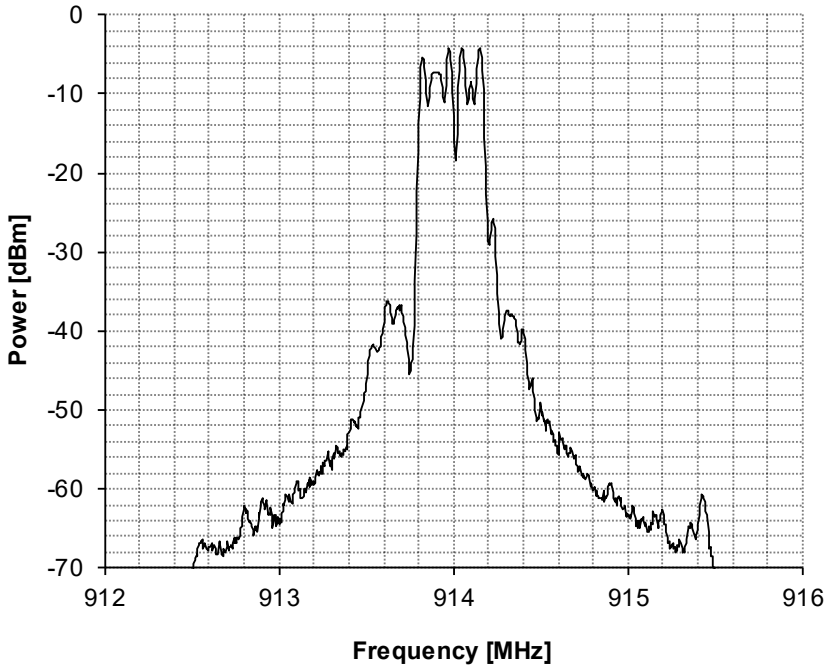


Figure 14-45. Spectrum of OQPSK-SIN-250

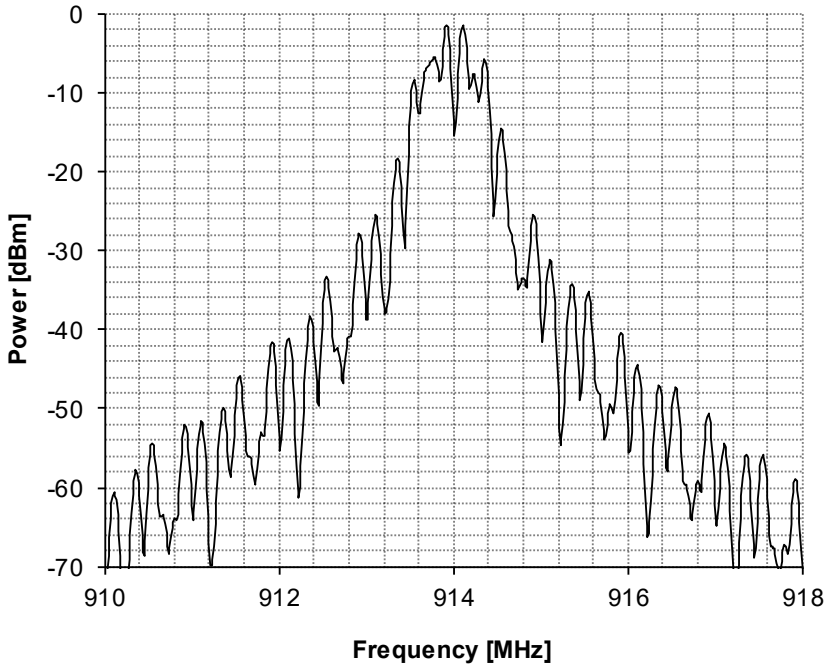
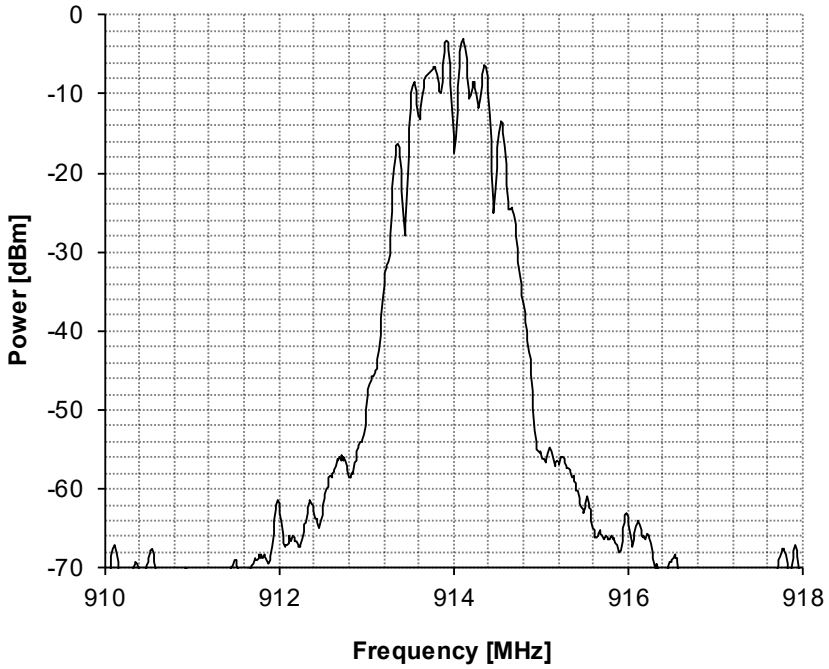


Figure 14-46. Spectrum of OQPSK-RC-250



The figures above illustrate typical spectra of the transmitted signals of the AT86RF212B and do not claim any limits.

Refer to the local authority bodies (FCC, ETSI, etc.) for further details about definition of power spectral density masks, definition of spurious emission, allowed modulation bandwidth, transmit power, and its limits.

Related Links

21. Continuous Transmission Test Mode

14.4.3.4 TX Output Power

The maximum output power of the transmitter is typically +5dBm in normal mode and +11dBm in boost mode. The TX output power can be set via the TX_PWR bits in the PHY_TX_PWR register (PHY_TX_PWR.TX_PWR). The output power of the transmitter can be controlled down to -25dBm with 1dB resolution.

To meet the spectral requirements of the European and Chinese bands, it is necessary to limit the TX power by appropriate setting of the TX_PWR and GC_PA bits in the PHY_TX_PWR register (PHY_TX_PWR.TX_PWR and PHY_TX_PWR.GC_PA), and the GC_TX_OFFS bits in the RF_CTRL_0 register (RF_CTRL_0.GC_TX_OFFS)..

Related Links

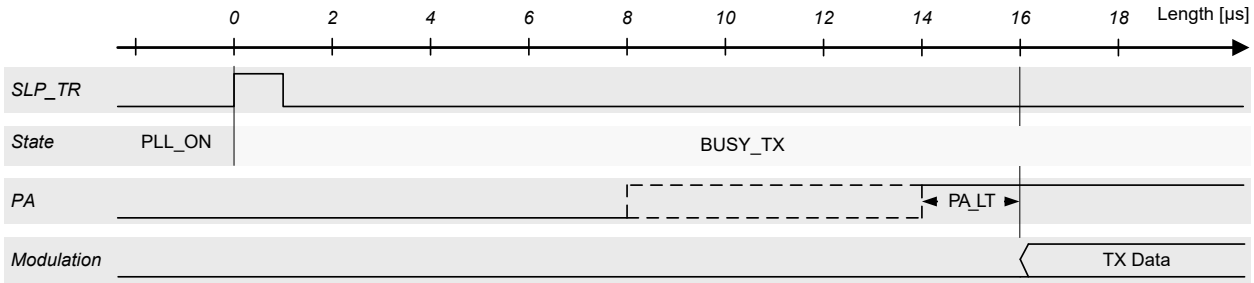
14.8.5 PHY_TX_PWR
14.8.22 RF_CTRL_0

14.4.3.5 TX Power Ramping

To optimize the output power spectral density (PSD), individual transmitter blocks are enabled sequentially. A transmit action is started by either the rising edge of pin 11 (SLP_TR) or by writing TX_START command to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). One symbol period later the data transmission begins. During this time period, the PLL settles to the frequency used for transmission. The PA is enabled prior to the data transmission start. This PA lead time can be adjusted with the PA_LT bits in the RF_CTRL_0 register (RF_CTRL_0.PA_LT). The PA is always enabled at the lowest gain value corresponding to GC_PA = 0. Then the PA gain is increased automatically to the value set by the GC_PA bits in the PHY_TX_PWR register (PHY_TX_PWR.GC_PA). After transmission is completed, TX power ramping down is performed in an inverse order.

The control signals associated with TX power ramping are shown in Figure 9-13. In this example, the transmission is initiated with the rising edge of pin 11 (SLP_TR). The radio transceiver state changes from PLL_ON to BUSY_TX.

Figure 14-47. TX Power Ramping Example (O-QPSK 250kb/s Mode)



Using an external RF front-end (refer to Section 11.4), it may be required to adjust the startup time of the external PA relative to the internal building blocks to optimize the overall PSD. This can be achieved using the RF_CTRL_0.PA_LT bits.

Related Links

14.8.22 RF_CTRL_0
14.8.2 TRX_STATE

14.4.4 Frame Buffer

The AT86RF212B contains a 128 byte dual port SRAM. One port is connected to the SPI interface, the other one to the internal transmitter and receiver modules. For data communication, both ports are independent and simultaneously accessible.

The Frame Buffer utilizes the SRAM address space 0x00 to 0x7F for RX and TX operation of the radio transceiver and can keep a single IEEE 802.15.4 RX or a single TX frame of maximum length at a time.

Frame Buffer access conflicts are indicated by an underrun interrupt IRQ_6 (TRX_UR).

Note: The IRQ_6 (TRX_UR) interrupt also occurs on the attempt to write frames longer than 127 octets to the Frame Buffer (overflow). In that case the content of the Frame Buffer cannot be guaranteed.

Frame Buffer access is only possible if the digital voltage regulator (DVREG) is turned on. This is valid in all device states except in SLEEP state. An access in P_ON state is possible if pin 17 (CLKM) provides the 1MHz master clock.

Related Links

14.1.3.2 Frame Buffer Access Mode

14.4.4.1 Data Management

Data in Frame Buffer (received data or data to be transmitted) remains valid as long as:

- No new frame or other data are written into the buffer over SPI
- No new frame is received (in any BUSY_RX state)
- No state change into SLEEP state is made
- No RESET took place

By default, there is no protection of the Frame Buffer against overwriting. Therefore, if a frame is received during Frame Buffer read access of a previously received frame, interrupt IRQ_6 (TRX_UR) is issued and the stored data might be overwritten.

Even so, the old frame data can be read, if the SPI data rate is higher than the effective over air data rate. For a data rate of 250kb/s, a minimum SPI clock rate of 1MHz is recommended. Finally, the microcontroller should check the transferred frame data integrity by an FCS check.

To protect the Frame Buffer content against being overwritten by newly incoming frames, the radio transceiver state should be changed to PLL_ON state after reception. This can be achieved by writing immediately the command PLL_ON to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD) after receiving the frame, indicated by IRQ_3 (TRX_END). Alternatively, Dynamic Frame Buffer Protection can be used to protect received frames against overwriting. Both procedures do not protect the Frame Buffer from overwriting by the microcontroller.

In Extended Operating Mode during TX_ARET operation the radio transceiver switches to receive state if an acknowledgement of a previously transmitted frame was requested. During this period, received frames are evaluated but not stored in the Frame Buffer. This allows the radio transceiver to wait for an acknowledgement frame and retry the frame transmission without writing the frame again.

A radio transceiver state change, except a transition to SLEEP state or a reset, does not affect the Frame Buffer content. If the radio transceiver is taken into SLEEP, the Frame Buffer is powered off and the stored data get lost.

Related Links

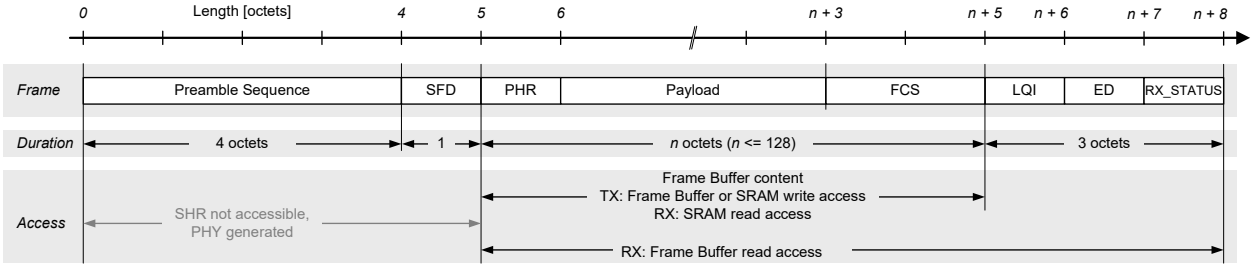
14.2.2 Extended Operationg Mode

14.2.2.4 TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry

14.4.4.2 User accessible Frame Content

The AT86RF212B supports an IEEE 802.15.4 compliant frame format as shown in the figure below.

Figure 14-48. AT86RF212B Frame Structure



A frame comprises two sections, the radio transceiver internally generated SHR field and the user accessible part stored in the Frame Buffer. The SHR contains the preamble and the SFD field. The variable frame section contains the PHR and the PSDU including the FCS, see Section 8.3.

To access the data, follow the procedures described in *Frame Check Sequence (FCS)*.

The frame length information (PHR field) and the PSDU are stored in the Frame Buffer. During frame reception, the link quality indicator (LQI) value, the energy detection (ED) value, and the status information (RX_STATUS) of a received frame are additionally stored. The radio transceiver appends these values to the frame data during Frame Buffer read access.

If the SRAM read access is used to read an RX frame, the frame length field (PHR) can be accessed at address zero. The SHR (except the SFD value used to generate the SHR) cannot be read by the microcontroller.

For frame transmission, the PHR and the PSDU needs to be stored in the Frame Buffer. The maximum Frame Buffer size supported by the radio transceiver is 128 bytes. If the TX_AUTO_CRC_ON bit in the TRX_CTRL_1 register (TRX_CTRL_1.TX_AUTO_CRC_ON) is set , the FCS field of the PSDU is replaced by the automatically calculated FCS during frame transmission.

To manipulate individual bytes of the Frame Buffer a SRAM write access can be used instead.

For non IEEE 802.15.4 compliant frames, the minimum frame length supported by the radio transceiver is one byte (Frame Length Field + one byte of data).

Related Links

14.1.3.2 Frame Buffer Access Mode
14.3.3 Frame Check Sequence (FCS)
14.3.5 Energy Detection
14.3.8 Link Quality Indication (LQI)
14.8.4 TRX_CTRL_1

14.4.4.3 Interrupt Handling

Access conflicts may occur when reading and writing data simultaneously at the two independent ports of the Frame Buffer, TX/RX BBP and SPI. These ports have their own address counter that points to the Frame Buffer's current address.

Access violations may cause data corruption and are indicated by IRQ_6 (TRX_UR) interrupt when using the Frame Buffer access mode. Note that access violations are not indicated when using the SRAM access mode.

While receiving a frame, first the data need to be stored in the Frame Buffer before reading it. This can be ensured by accessing the Frame Buffer at least eight symbols (BPSK) or two symbols (O-QPSK) after interrupt IRQ_2 (RX_START). When reading the frame data continuously, the SPI data rate shall be lower than the current TRX bit rate to ensure no underrun interrupt occurs. To avoid access conflicts and to simplify the Frame Buffer read access, Frame Buffer Empty indication may be used.

When writing data to the Frame Buffer during frame transmission, the SPI data rate shall be higher than the PHY data rate avoiding underrun. The first byte of the PSDU data must be available in the Frame Buffer before SFD transmission is complete, which takes 41 symbol periods for BPSK (one symbol PA ramp up + 40 symbols SHR) and 11 symbol periods for O-QPSK (one symbol PA ramp up + 10 symbols SHR) from the rising edge of pin 11 (SLP_TR) (see Figure 14-20).

Notes:

1. Interrupt IRQ_6 (TRX_UR) is valid two octets after IRQ_2 (RX_START).
2. If a Frame Buffer read access is not finished until a new frame is received, an IRQ_6 (TRX_UR) interrupt occurs. Nevertheless the old frame data can be read, if the SPI data rate is higher than the effective PHY data rate. A minimum SPI clock rate of 1MHz is recommended in this case. Finally, the microcontroller should check the integrity of the transferred frame data by calculating the FCS.
3. When writing data to the Frame Buffer during frame transmission, the SPI data rate shall be higher than the PHY data rate to ensure no under run interrupt. The first byte of the PSDU data must be available in the Frame Buffer before SFD transmission is complete.

Related Links

14.2.1.3 Interrupt Handling

14.4.5 Voltage Regulators (AVREG, DVREG)

The main features of the Voltage Regulator blocks are:

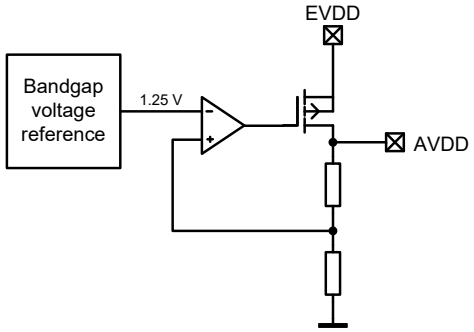
- Bandgap stabilized 1.8V supply for analog and digital domain
- Low dropout (LDO) voltage regulator
- AVREG/DVREG can be disabled when an external regulated voltage is supplied to AVDD/DVDD pin

14.4.5.1 Overview

The internal voltage regulators supply a stabilized voltage to the AT86RF212B. The AVREG provides the regulated 1.8V supply voltage for the analog domain and the DVREG supplies the 1.8V supply voltage for the digital domain.

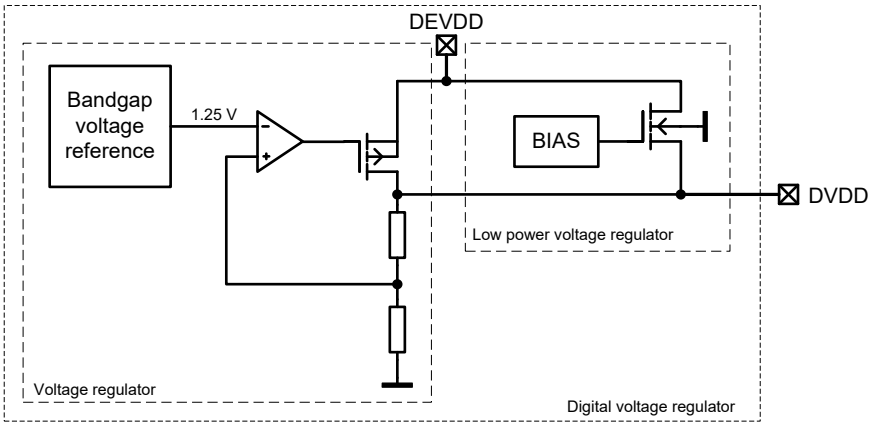
A simplified schematic of the internal analog voltage regulator is shown in the figure below.

Figure 14-49. Simplified Schematic of AVREG



A simplified schematic of the internal digital voltage regulator is shown in the figure below.

Figure 14-50. Simplified Schematic of DVREG



The block “Low power voltage regulator” within the “Digital voltage regulator” maintains the DVDD supply voltage at 1.5V (typical) when the AT86RF212B voltage regulator is disabled in sleep mode. All configuration register values are stored.

The low power voltage regulator is always enabled. Therefore, its bias current contributes to the leakage current in sleep mode with about 100nA (typical).

The voltage regulators (AVREG, DVREG) require bypass capacitors for stable operation. The value of the bypass capacitors determine the settling time of the voltage regulators. The bypass capacitors shall be placed as close as possible to the pins and shall be connected to ground with the shortest possible traces.

14.4.5.2 Configuration

The voltage regulators can be configured by the VREG_CTRL register.

It is recommended to use the internal regulators, but it is also possible to supply the low voltage domains by an external voltage supply. For this configuration, the internal regulators need to be switched off by setting the register bits to the values AVREG_EXT = 1 and DVREG_EXT = 1. A regulated external supply voltage of 1.8V needs to be connected to the pins 13, 14 (DVDD) and pin 29 (AVDD). Even if DVDD and AVDD are connected to an external supply, it is required to connect VDD to an external supply. When providing the external supply, ensure a sufficiently long stabilization time before interacting with the AT86RF212B.

Disabling the internal regulators increases total SLEEP current for DVDD/DEVDD to 800nA/150nA. Note that the combined nominal current for DEVDD is only 200nA with internal regulators enabled.

Related Links

14.8.16 VREG_CTRL

14.4.5.3 Data Interpretation

The status bits AVDD_OK = 1 and DVDD_OK = 1 in the VREG_CTRL register indicate an enabled and stable internal supply voltage. Reading value zero indicates a disabled or internal supply voltage not settled to the final value. Setting AVREG_EXT = 1 and DVREG_EXT = 1 forces the signals AVDD_OK and DVDD_OK to one.

Related Links

14.8.16 VREG_CTRL

14.4.6 Battery Monitor (BATMON)

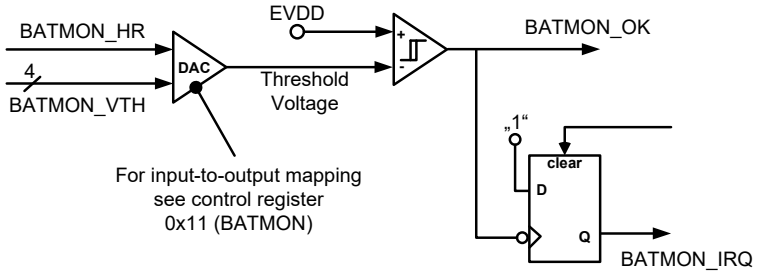
The main features of the battery monitor are:

- Configurable voltage reference threshold from 1.70V to 3.675V
- Interrupt on low - supply voltage condition
- Continuous BATMON status monitor as a register flag

14.4.6.1 Overview

The AT86RF212B battery monitor (BATMON) detects and flags a low external supply voltage level. provided on pin 28 (EVDD). The external voltage supply pin 28 (EVDD) is continuosly compared with the internal threshold voltage to detect a low voltage supply level. In this case BATMON_IRQ is triggered and BATMON_OK flag is cleared to indicate undervoltage condition,.

Figure 14-51. Simplified Schematic of BATMON



14.4.6.2 Configuration

The BATMON can be configured using the BATMON register. The BATMON_VTH bits sets the threshold voltage. It is configurable with a resolution of 75mV in the upper voltage range (BATMON_HR = 1) and with a resolution of 50mV in the lower voltage range (BATMON_HR = 0).

Related Links

14.8.17 BATMON

14.4.6.3 Data Interpretation

The BATMON_OK bit in the BATMON register (BATMON.BATMON_OK) monitors the current value of the battery voltage:

- If BATMON_OK = 0, the battery voltage is lower than the threshold voltage
- If BATMON_OK = 1, the battery voltage is higher than the threshold voltage

After setting a new threshold, the value BATMON_OK should be read out to verify the current supply voltage value.

Note: The battery monitor is inactive during P_ON, and SLEEP states, see the TRX_STATUS bits in the TRX_STATUS register (TRX_STATUS.TRX_STATUS).

Related Links

14.8.17 BATMON

14.8.1 TRX_STATUS

14.4.6.4 Interrupt Handling

A supply voltage drop below the configured threshold value is indicated by an interrupt IRQ_7 (BAT_LOW).

Note: The AT86RF212B IRQ_7 (BAT_LOW) interrupt is issued only if the BATMON.BATMON_OK bit changes from one to zero.

IRQ_7 (BAT_LOW) interrupt is not generated under following conditions:

- The battery voltage remained below 1.8V threshold value on power-on (the BATMON.BATMON_OK bit was never one), or
- A new threshold is set, which is still above the current supply voltage (the BATMON.BATMON_OK bit remains zero).

When the battery voltage is close to the programmed threshold voltage, noise or temporary voltage drops may generate unwanted interrupts. To avoid this:

- Disable the IRQ_7 (BAT_LOW) in the IRQ_MASK register and treat the battery as empty, or
- Set a lower threshold value.

Related Links

14.1.7 Interrupt Logic
14.8.17 BATMON
14.8.14 IRQ_MASK

14.4.7 Crystal Oscillator (XOSC) and Clock Output (CLKM)

The main crystal oscillator features are:

- 16MHz amplitude-controlled crystal oscillator
- Fast settling time after leaving SLEEP state
- Configurable trimming capacitance array
- Configurable clock output (CLKM)

14.4.7.1 Overview

The crystal oscillator generates the reference frequency for the AT86RF212B. All other internally generated frequencies of the radio transceiver are derived from this frequency. Therefore, the overall system performance is mainly determined by the accuracy of crystal reference frequency. The external components of the crystal oscillator should be selected carefully and the related board layout should be done with caution.

The XOSC_CTRL register provides access to the control signals of the oscillator. Two operating modes are supported. It is recommended to use the integrated oscillator setup. Alternatively, a reference frequency can be fed to the internal circuitry by using an external clock reference.

Related Links

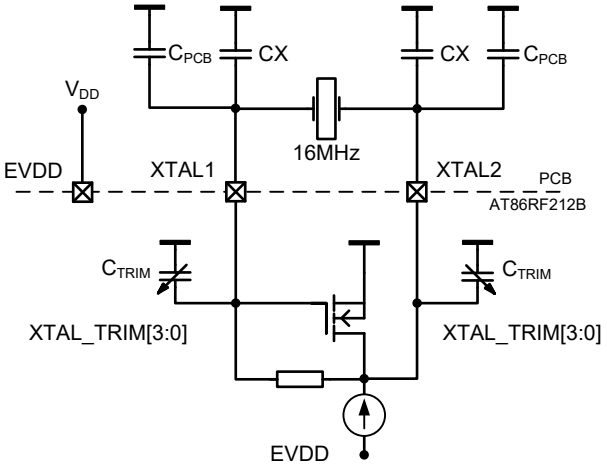
14.4.7.2 Integrated Oscillator Setup
14.4.7.4 Master Clock Signal Output (CLKM)
14.8.18 XOSC_CTRL

14.4.7.2 Integrated Oscillator Setup

Using the internal oscillator, the oscillation frequency depends on the load capacitance between the crystal pin 26 (XTAL1) and pin 25 (XTAL2). The total load capacitance C_L must be equal to the specified load capacitance of the crystal itself. It consists of the external capacitors CX and parasitic capacitances connected to the XTAL nodes.

The figure below shows all parasitic capacitances, such as PCB stray capacitances and the pin input capacitance, summarized to C_{PAR}.

Figure 14-52. Simplified XOSC Schematic with External Components



Additional internal trimming capacitors C_{TRIM} are available. Any value in the range from 0pF to 4.5pF with a 0.3pF resolution is selectable using the XTAL_TRIM bits in the XOSC_CTRL register (XOSC_CTRL.XTAL_TRIM). To calculate the total load capacitance, the following formula can be used

$C_L[pF] = 0.5 \times (CX[pF] + C_{TRIM}[pF] + C_{PAR}[pF])$.

The AT86RF212B trimming capacitors provide the possibility of reducing frequency deviations caused by production process variations or by external components tolerances. Note that the oscillation frequency can only be reduced by increasing the trimming capacitance. The frequency deviation caused by one step of C_{TRIM} decreases with increasing crystal load capacitor values.

An amplitude control circuit is included to ensure stable operation under different operating conditions and for different crystal types. Enabling the crystal oscillator in P_ON state and after leaving SLEEP state causes a slightly higher current during the amplitude build-up phase to guarantee a short start-up time. At stable operation, the current is reduced to the amount necessary for a robust operation. This also keeps the drive level of the crystal low.

Generally, crystals with a higher load capacitance are less sensitive to parasitic pulling effects caused by external component variations or by variations of board and circuit parasitic. On the other hand, a larger crystal load capacitance results in a longer start-up time and a higher steady state current consumption.

Related Links

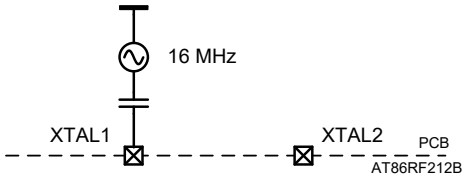
14.8.18 XOSC_CTRL

14.4.7.3 External Reference Frequency Setup

When using an external reference frequency, the signal must be connected to pin 26 (XTAL1) as indicated in the figure below and the XTAL_MODE bits in the XOSC_CTRL register (XOSC_CTRL.XTAL_MODE) need to be set to the external oscillator mode for power saving reasons. The oscillation peak-to-peak amplitude shall be between 100mV and 500mV, the optimum range is between 400mV and 500mV. Pin 25 (XTAL2) should not be wired. It is possible, among other waveforms, to use sine and square wave signals.

Note: The quality of the external reference (that is phase noise) determines the system performance.

Figure 14-53. Setup for Using an External Frequency Reference



Related Links

14.8.18 XOSC_CTRL

14.4.7.4 Master Clock Signal Output (CLKM)

The generated reference clock signal can be fed into a microcontroller using pin 17 (CLKM). The internal 16MHz raw clock can be divided by an internal prescaler. Thus, clock frequencies of 16MHz, 8MHz, 4MHz, 2MHz, 1MHz, 250kHz, or the current SHR symbol rate frequency can be supplied by pin 17 (CLKM).

The CLKM frequency update scheme and pin driver strength is configurable using the TRX_CTRL_0 register. There are two possibilities how a CLKM frequency change gets effective:

If CLKM_SHA_SEL = 0 and/or CLKM_CTRL = 0, changing the CLKM_CTRL bits in the TRX_CTRL_0 register (TRX_CTRL_0.CLKM_CTRL) will immediately affects a glitch free CLKM clock rate change.

Otherwise (CLKM_SHA_SEL = 1 and CLKM_CTRL > 0 before changing the CLKM_CTRL bits), the new clock rate is supplied when leaving the SLEEP state the next time.

To reduce power consumption and spurious emissions, it is recommended to turn off the CLKM clock when not in use or to reduce its driver strength to a minimum.

Note:
During reset procedure the CLKM_CTRL bits are shadowed. Although the clock setting of CLKM remains after reset, a read access to register bits CLKM_CTRL delivers the reset value one. For that reason, it is recommended to write the previous configuration (before reset) to register bits CLKM_CTRL (after reset) to align the radio transceiver behavior and register configuration. Otherwise, the CLKM clock rate is set back to the reset value (1MHz) after the next SLEEP cycle.

For example, if the CLKM clock rate is configured to 16MHz, the CLKM clock rate remains at 16MHz after a reset, however, the register bits CLKM_CTRL are set back to one. Since CLKM_SHA_SEL reset value is one, the CLKM clock rate changes to 1MHz after the next SLEEP cycle if the CLKM_CTRL setting is not updated.

Related Links

14.2.1.4.5 Reset Procedure
14.8.3 TRX_CTRL_0

14.4.7.5 Clock Jitter

The AT86RF212B provides receiver sensitivities up to -110dBm. Detection of such small RF signals requires very clean scenarios with respect to noise and interference. Harmonics of digital signals may degrade the performance if they interfere with the wanted RF signal. A small clock jitter of digital signals can spread harmonics over a wider frequency range, thus reducing the power of certain spectral lines. The AT86RF212B provides such a clock jitter as an optional feature. The jitter module is working for the receiver part and all I/O signals, for example CLKM if enabled. The transmitter part and RF frequency generation are not influenced.

14.4.8 Frequency Synthesizer (PLL)

The main PLL features are:

- Generate RX/TX frequencies for all supported channels
- Autonomous calibration loops for stable operation within the operating range
- Two PLL interrupts for status indication
- Fast PLL settling to support frequency hopping

14.4.8.1 Overview

The PLL generates the RF frequencies for the AT86RF212B. During receive and transmit operations, the frequency synthesizer operates as a local oscillator. The frequency synthesizer is implemented as a fractional-N PLL with analog compensation of the fractional phase error. The voltage-controlled oscillator (VCO) is running at double of the RF frequency.

Two calibration loops ensure correct PLL functionality within the specified operating limits.

14.4.8.2 RF Channel Selection

The PLL is designed to support:

- One channel in the European SRD band from 863MHz to 870MHz at 868.3MHz according to IEEE 802.15.4 (channel k = 0)
- 10 channels in the North American ISM band from 902MHz to 928MHz with a channel spacing of 2MHz according to IEEE 802.15.4. The center frequency of these channels is defined as:

F_C[MHz] = 906[MHz] + 2[MHz] x (k – 1), for k = 1, 2, ..., 10

where k is the channel number.

- Four channels in the Chinese WPAN band from 779MHz to 787MHz with a channel spacing of 2MHz according to IEEE 802.15.4c-2009 and IEEE 802.15.4-2011. Center frequencies are 780MHz, 782MHz, 784MHz, and 786MHz.

Additionally, the PLL supports all frequencies from 769MHz to 935MHz with 1MHz frequency spacing and four bands with 100kHz spacing from 769.0MHz to 794.5MHz, 857.0MHz to 882.5MHz, and 902.0MHz to 928.5MHz. The frequency is selected by the CC_BAND bits in the CC_CTRL_1 register (CC_CTRL_1.CC_BAND) and CC_NUMBER bits in the CC_CTRL_0 register (CC_CTRL_0.CC_CTRL_0).

The table below shows the settings of CC_BAND and CC_NUMBER.

Table 14-33. Frequency Bands and Numbers

CC_BAND	CC_NUMBER	Description
0	Not used	European and North American channels according to IEEE 802.15.4; Frequency selected by the CHANNEL bits in the PHY_CC_CCA register (PHY_CC_CCA.CHANNEL)
1	0x00 – 0xFF	769.0MHz – 794.5MHz Fc [MHz] = 769.0[MHz] + 0.1[MHz] x CC_NUMBER
2	0x00 – 0xFF	857.0MHz – 882.5MHz Fc [MHz] = 857.0[MHz] + 0.1[MHz] x CC_NUMBER
3	0x00 – 0xFF	903.0MHz – 928.5MHz Fc [MHz] = 903.0[MHz] + 0.1[MHz] x CC_NUMBER
4	0x00 – 0x5E	769MHz – 863MHz Fc [MHz] = 769[MHz] + 1[MHz] x CC_NUMBER
5	0x00 – 0x66	833MHz – 935MHz Fc [MHz] = 833[MHz] + 1[MHz] x CC_NUMBER
6	0x00 – 0xFF	902.0MHz – 927.5MHz Fc [MHz] = 902.0[MHz] + 0.1[MHz] x CC_NUMBER
7	0x00 – 0xFF	Reserved

Related Links

14.8.19 CC_CTRL_0
14.8.20 CC_CTRL_1
14.8.8 PHY_CC_CCA

14.4.8.3 PLL Settling Time and Frequency Agility

When the PLL is enabled during state transition from TRX_OFF to PLL_ON or RX_ON, the settling time is typically t_{TR4} = 170µs, including PLL self calibration. A lock of the PLL is indicated with an interrupt IRQ_0 (PLL_LOCK).

Switching between channels within a frequency band in PLL_ON or RX_ON states is typically done within t_{PLL_SW} = 11µs. This makes the radio transceiver highly suitable for frequency hopping applications.

The PLL frequency in PLL_ON and receive states is 1MHz below the PLL frequency in transmit states. When starting the transmit procedure, the PLL frequency is changed to the transmit frequency within a period of t_{RX_TX} = 16µs before really starting the transmission. After the transmission, the PLL settles back to the receive frequency within a period of t_{TX_RX} = 32µs. This frequency step does not generate an interrupt IRQ_0 (PLL_LOCK) or IRQ_1 (PLL_UNLOCK) within these periods.

Related Links

14.2.1.4.6 State Transition Timing Summary

14.4.8.4 Calibration Loops

14.4.8.4 Calibration Loops

Due to variation of temperature, supply voltage and part-to-part variations of the radio transceiver the VCO characteristics may vary.

To ensure a stable operation, two automated control loops are implemented:

- Center Frequency (CF) tuning
- Delay Cell (DCU) calibration

Both calibration loops are initiated automatically when the PLL is enabled during state transition from TRX_OFF to PLL_ON or RX_ON state. Additionally, both calibration loops are initiated when the PLL changes to a different frequency setting.

If the PLL operates for a long time on the same channel, for example more than five minutes, or the operating temperature changes significantly, it is recommended to initiate the calibration loops manually.

Both calibration loops can be initiated manually by SPI command. To start the calibration, the device should be in state PLL_ON. The center frequency calibration can be initiated by setting the PLL_CF_START bit in the PLL_CF register to '1' (PLL_CF.PLL_CF_START=1). The calibration loop is completed when the IRQ_0 (PLL_LOCK) occurs, if enabled. The duration of the center frequency calibration loop depends on the difference between the current CF value and the final CF value. During the calibration, the CF value is incremented or decremented. Each step takes t_{PLL_CF} = 8µs. The minimum time is 8µs; the maximum time is 270µs. The recommended procedure to start the center frequency calibration is to read the register 0x1A (PLL_CF), to set the PLL_CF_START register bit to one, and to write the value back to the register.

The delay cell calibration can be initiated by setting the PLL_DCU_START bit in the PLL_DCU register (PLL_DCU.PLL_DCU_START) to '1'. The delay time of the programmable delay unit is adjusted to the correct value. The calibration works as successive approximation and is independent of the values in the PLL_DCU register. The duration of the calibration is t_{PLL_DCU} = 10µs.

During both calibration processes, no correct receive or transmit operation is possible. The recommended state for the calibration is therefore PLL_ON, but calibration is not blocked at receive or transmit states.

Both calibrations can be executed concurrently.

Related Links

14.8.25 PLL_CF

14.8.26 PLL_DCU

14.4.8.5 Interrupt Handling

Two different interrupts indicate the PLL status (refer to the IRQ_STATUS register). IRQ_0 (PLL_LOCK) indicates that the PLL has locked. IRQ_1 (PLL_UNLOCK) interrupt indicates an unexpected unlock condition. A PLL_LOCK interrupt clears any preceding PLL_UNLOCK interrupt automatically and vice versa.

An IRQ_0 (PLL_LOCK) interrupt is supposed to occur in the following situations:

- State change from TRX_OFF to PLL_ON / RX_ON
- Frequency setting change in states PLL_ON / RX_ON
- A manually started center frequency calibration has been completed

All other PLL_LOCK interrupt events indicate that the PLL locked again after a prior unlock happened.

An IRQ_1 (PLL_UNLOCK) interrupt occurs in the following situations:

- A manually initiated center frequency calibration in states PLL_ON / (RX_ON)
- Frequency setting change in states PLL_ON / RX_ON

Any other occurrences of IRQ_1 (PLL_UNLOCK) indicate erroneous behavior and require checking of the actual device status.

PLL_LOCK and PLL_UNLOCK affect the behavior of the transceiver:

In states BUSY_TX and BUSY_TX_ARET the transmission is stopped and the transceiver returns into state PLL_ON. During BUSY_RX and BUSY_RX_AACK, the transceiver returns to state RX_ON and RX_AACK_ON, respectively, once the PLL has locked.

Notes:

1. An AT86RF212B interrupt IRQ_0 (PLL_LOCK) clears any preceding IRQ_1 (PLL_UNLOCK) interrupt automatically and vice versa.
2. The state transition from BUSY_TX / BUSY_TX_ARET to PLL_ON / TX_ARET_ON after successful transmission does not generate an IRQ_0 (PLL_LOCK) within the settling period.

Related Links

14.8.15 IRQ_STATUS

14.4.9 Automatic Filter Tuning (FTN)

14.4.9.1 Overview

The Automatic Filter Tuning (FTN) is incorporated to compensate device tolerances for temperature, supply voltage variations as well as part-to-part variations of the radio transceiver. The filter-tuning result is used to correct the analog baseband filter transfer function and the PLL loop-filter time constant.

An FTN calibration cycle is initiated automatically when entering the TRX_OFF state from the P_ON, SLEEP, or RESET state.

Although receiver and transmitter are very robust against these variations, it is recommended to initiate the FTN manually if the radio transceiver does not use the SLEEP state. If necessary, a calibration cycle is to be initiated in states TRX_OFF, PLL_ON or RX_ON. This applies in particular for the High Data Rate Modes with a much higher sensitivity against Band-Pass Filter (BPF) transfer function variations. The recommended calibration interval is five minutes or less, if the AT86RF212B operates always in an active state (PLL_ON, TX_ARET_ON, RX_ON, and RX_AACK_ON).

Related Links

4.3 Transceiver Circuit Description
14.4.1.4 Proprietary High Data Rate Modes

14.5 Radio Transceiver Usage

This section describes basic procedures to receive and transmit frames using the AT86RF212B. For a detailed programming description refer to reference [11].

14.5.1 Frame Receive Procedure

A frame reception comprises of two actions: The transceiver listens for, receives, and demodulates the frame to the Frame Buffer and signals the reception to the microcontroller. After or during that process, the microcontroller can read the available frame data from the Frame Buffer via the SPI interface.

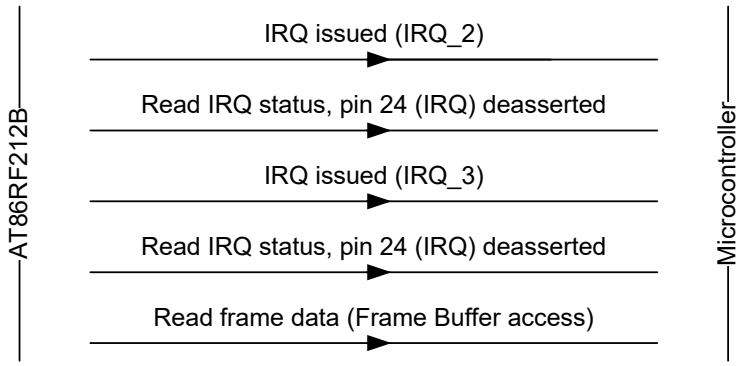
While being in state RX_ON or RX_AACK_ON, the radio transceiver searches for incoming frames with the selected modulation scheme and data rate on the selected channel. Assuming the appropriate interrupts are enabled, the detection of a frame is indicated by interrupt IRQ_2 (RX_START). When the frame reception is completed, interrupt IRQ_3 (TRX_END) is issued.

Different Frame Buffer read access scenarios are recommended for:

- Non-time critical applications:
  - Read access starts after IRQ_3 (TRX_END)
- Time-critical applications:
  - Read access starts after IRQ_2 (RX_START)

For non-time-critical operations, it is recommended to wait for interrupt IRQ_3 (TRX_END) before starting a Frame Buffer read access. The figure below illustrates the frame receive procedure using IRQ_3 (TRX_END).

Figure 14-54. Transactions between AT86RF212B and Microcontroller during Receive



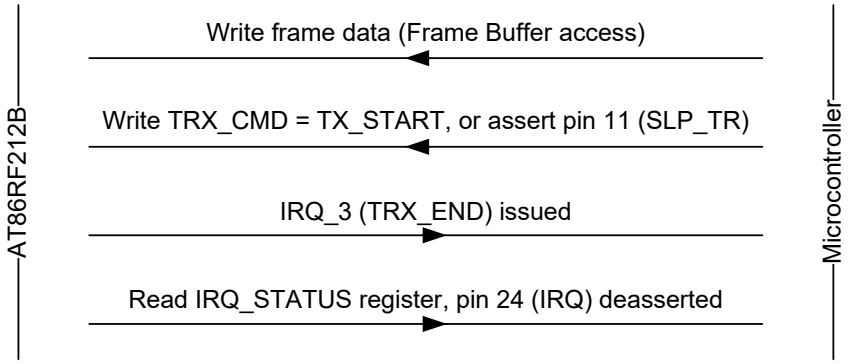
Critical protocol timing could require starting the Frame Buffer read access after interrupt IRQ_2 (RX_START). The first byte of the frame data can be read 32µs after the IRQ_2 (RX_START) interrupt. The microcontroller must ensure to read slower than the frame is received. Otherwise a Frame Buffer under run occurs, IRQ_6 (TRX_UR) is issued, and the frame data may be not valid. To avoid this, the Frame Buffer read access can be controlled by using a Frame Buffer Empty Indicator.

14.5.2 Frame Transmit Procedure

A frame transmission comprises of two actions, a write to Frame Buffer and the transmission of its contents. Both actions can be run in parallel if required by critical protocol timing.

The figure below illustrates the AT86RF212B frame transmit procedure, when writing and transmitting the frame consecutively. After a Frame Buffer write access, the frame transmission is initiated by asserting pin 11 (SLP_TR) or writing command TX_START to the TRX_CMD bits in the TRX_STATE register (TRX_STATE.TRX_CMD). The transceiver must be either in PLL_ON state for basic operating mode or TX_ARET_ON state for extended operating mode. The completion of the transaction is indicated by interrupt IRQ_3 (TRX_END).

Figure 14-55. Transaction between AT86RF212B and Microcontroller during Transmit

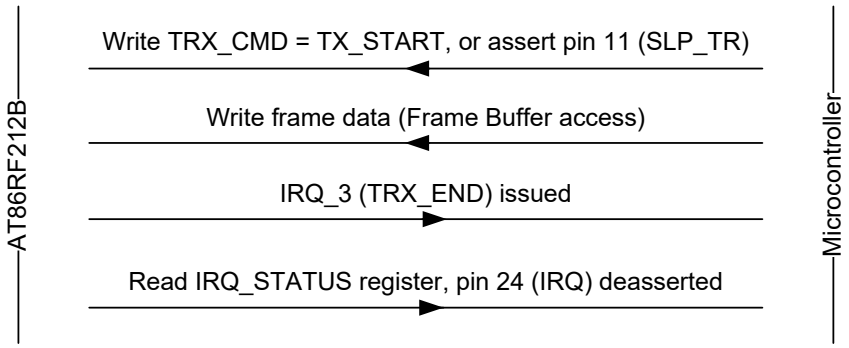


Alternatively for time critical applications when the frame start transmission time needs to be minimized, a frame transmission task can be started first. Then it can be followed by the Frame Buffer write access event (populating PSDU data). This way the data to be transmitted is needs to be written in the transmit frame buffer as the transceiver initializes and begins SHR transmission.

By initiating a transmission, either by asserting pin 11 (SLP_TR) or writing a TX_START command to the TRX_CMD bits, the radio transceiver starts transmitting the SHR, which is internally generated.

Front end initialization takes one symbol period to settle PLL and ramp up the PA. SHR transmission takes another 40 symbol periods for BPSK or 10 symbol periods delay for O-QPSK. By this time the PHR must be available in the Frame Buffer. Furthermore, the SPI data rate must be higher than the PHY data rate to avoid a Frame Buffer underrun, which is indicated by IRQ_6 (TRX_UR).

Figure 14-56. Time Optimized Frame Transmit Procedure



Related Links

14.4.1 Physical Layer Modes
14.8.2 TRX_STATE

14.6 Extended Feature Set

14.6.1 Security Module (AES)

14.6.1.1 Overview

The security module is based on an AES-128 core according to FIPS197 standard, refer to [10]. The security module works independently of other building blocks of the AT86RF212B. Encryption and decryption can be performed in parallel with a frame transmission or reception.

The control of the security block is implemented as an SRAM access to address space 0x82 to 0x94. A Fast SRAM access mode allows for simultaneous new data writes and reads of processed data within the same SPI transfer. This access procedure is used to reduce the turnaround time for ECB and CBC modes.

In addition, the security module contains another 128-bit register to store the initial key used for security operations. This initial key is not modified by the security module.

Related Links

14.6.1.5 Data Transfer – Fast SRAM Access
23. References

14.6.1.2 Security Module Preparation

The use of the security module requires a configuration of the security engine before starting a security operation. The following steps are required:

Table 14-34. AES Engine Configuration Steps

Step	Description	Description
1	Key Setup	Write encryption or decryption key to SRAM
2	AES mode	Select AES mode: ECB or CBC Select encryption or decryption
3	Write Data	Write plaintext or cipher text to SRAM
4	Start operation	Start AES operation
5	Read Data	Read cipher text or plaintext from SRAM

Before starting any security operation, a key must be written to the security engine. The key set up requires the configuration of the AES engine KEY mode using the AES_MODE bits in the AES_CTRL register (AES_CTRL.AES_MODE).

The following step selects the AES mode, either electronic code book (ECB) or cipher block chaining (CBC). Further, encryption or decryption must be selected with the AES_DIR bit in the AES_CTRL register (AES_CTRL.AES_DIR).

After this, the 128-bit plain text or cipher text data has to be provided to the AES hardware engine. The data uses the SRAM address range 0x84 – 0x93.

An encryption or decryption is initiated with bit AES_REQUEST = 1 (either in the SRAM address 0x83 AES_CTRL, or the mirrored version SRAM address 0x94 AES_CTRL_MIRROR).

The AES module control registers are only accessible using SRAM read and write accesses on address space 0x82 to 0x94. Configuring the AES mode, providing the data, and starting a decryption or encryption operation can be combined in a single SRAM access.

Notes:

1. No additional register access is required to operate the security block.
2. Access to the security block is not possible while the radio transceiver is in SLEEP, or RESET state.
3. All configurations of the security module, the SRAM content, and keys are reset during RESET state.
4. A read or write access to the AES_CTRL register during AES operation terminates the current processing.

Related Links

14.6.1.5 Data Transfer – Fast SRAM Access
14.6.1.4.2 Cipher Block Chaining (CBC)
14.6.1.4.1 Electronic Code Book (ECB)
14.6.1.3 Security Key Setup
14.6.1.4 Security Operation Modes

14.6.1.3 Security Key Setup

The setup of the key is prepared by setting the AES_MODE bit in the AES_CTRL register (AES_CTRL-AES_MODE) to '1'. Afterwards the 128-bit key must be written to the AES_KEY registers. It is recommended to combine the setting of AES_CTRL register and the 128-bit key transfer using only one SRAM access starting from address 0x83.

The address space for the 128-bit key and 128-bit data is identical from programming point of view. However, both use different pages which are selected by the AES_MODE bits in the AES_CTRL register (AES_CTRL.AEC_MODE) before storing the data.

A read access to the AES_KEY registers (0x84 – 0x93) returns the last round key of the preceding security operation. After an ECB encryption operation, this is the key that is required for the corresponding ECB decryption operation. However, the initial AES key, written to the security module in advance of an AES run (step one ithe *AES Engine Configuration Steps* table) is not modified during the AES operation. This initial key is used for the next AES run even it cannot be read from AES_KEY.

Note: ECB decryption is not required for IEEE 802.15.4 or ZigBee security processing. The AT86RF212B provides this functionality as an additional feature.

Related Links

14.6.1.2 Security Module Preparation

14.6.1.4 Security Operation Modes

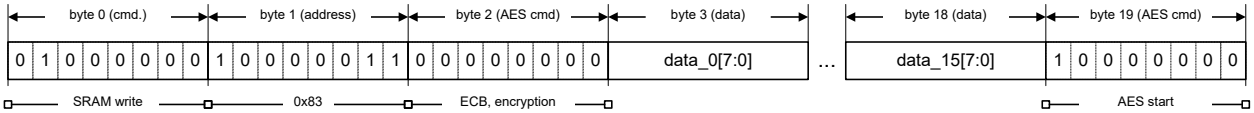
14.6.1.4.1 Electronic Code Book (ECB)

ECB is the basic operating mode of the security module. After setting up the initial AES key, AES_CTRL.AES_MODE = 0 sets up ECB mode. The AES_DIR bits in the AES_CTRL register (AES_CTRL.AES_DIR) selects the direction, either encryption or decryption. The data to be processed has to be written to SRAM addresses 0x84 through 0x93 (AES_STATE registers).

An example for a programming sequence is shown in the figure below. This example assumes a suitable key has been loaded before.

A security operation can be started within one SRAM access by appending the start command AES_REQUEST = 1 (register 0x94, AES_CTRL_MIRROR) to the SPI sequence. Register AES_CTRL_MIRROR is a mirrored version of register 0x83 (AES_CTRL).

Figure 14-57. ECB Programming SPI Sequence – Encryption



Summarizing, the following steps are required to perform a security operation using only one AT86RF212B SPI access:

1. Configure SPI access
   1.1. SRAM write
   1.2. Start address 0x83
2. Configure AES operation
   2.1. address 0x83: select ECB mode, direction
3. Write 128-bit data block
   3.1. addresses 0x84 – 0x93: either plain or ciphertext
4. Start AES operation
   4.1. address 0x94: start AES operation, ECB mode

This sequence is recommended because the security operation is configured and started within one SPI transaction.

The ECB encryption and decryption operation is illustrated in the figures below.

Figure 14-58. ECB Mode – Encryption

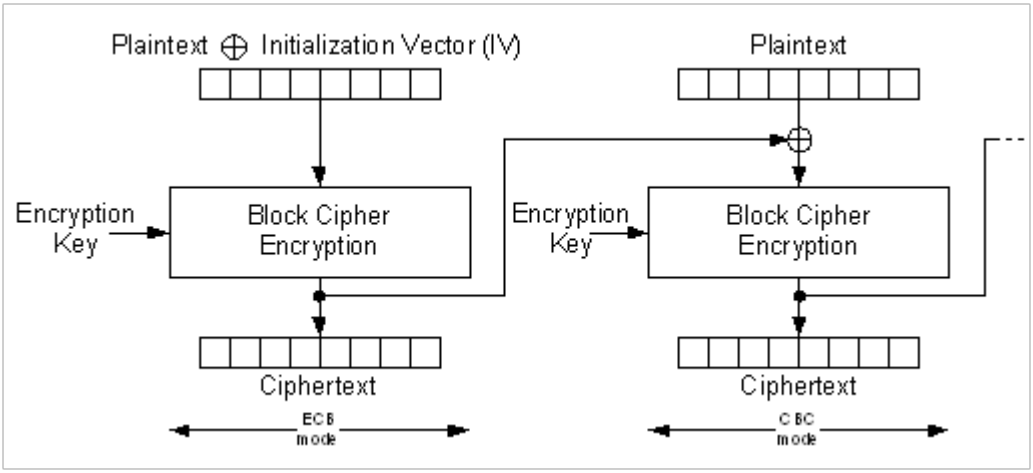
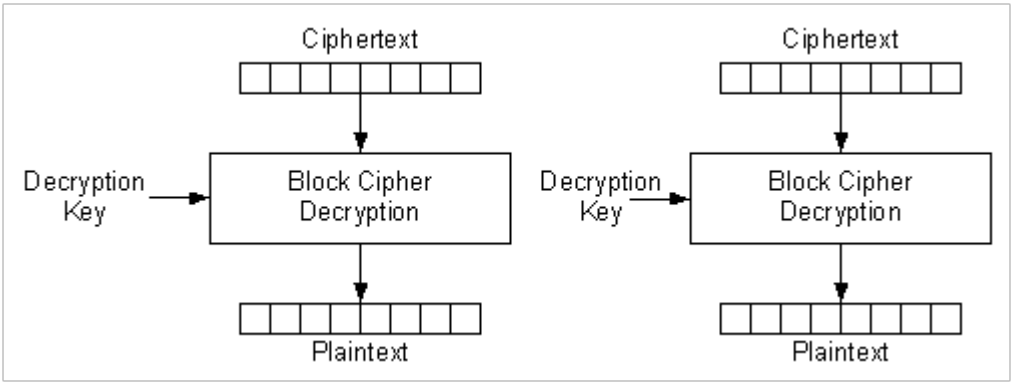


Figure 14-59. ECB Mode – Decryption



When decrypting, due to the nature of AES algorithm, the initial key to be used is not the same as the one used for encryption, but rather the last round key instead. This last round key is the content of the key address space stored after running one full encryption cycle, and must be saved for decryption. If the decryption key has not been saved, it has to be recomputed by first running a dummy encryption (of an arbitrary plaintext) using the original encryption key, then fetching the resulting round key from the key memory, and writing it back into the key memory as the decryption key.

ECB decryption is not used by either IEEE 802.15.4 or ZigBee frame security. Both of these standards do not directly encrypt the payload, but rather a nonce instead, and protect the payload by applying an XOR operation between the resulting (AES-) cipher text and the original payload. As the nonce is the same for encryption and decryption only ECB encryption is required. Decryption is performed by XORing the received cipher text with its own encryption result respectively, which results in the original plaintext payload upon success.

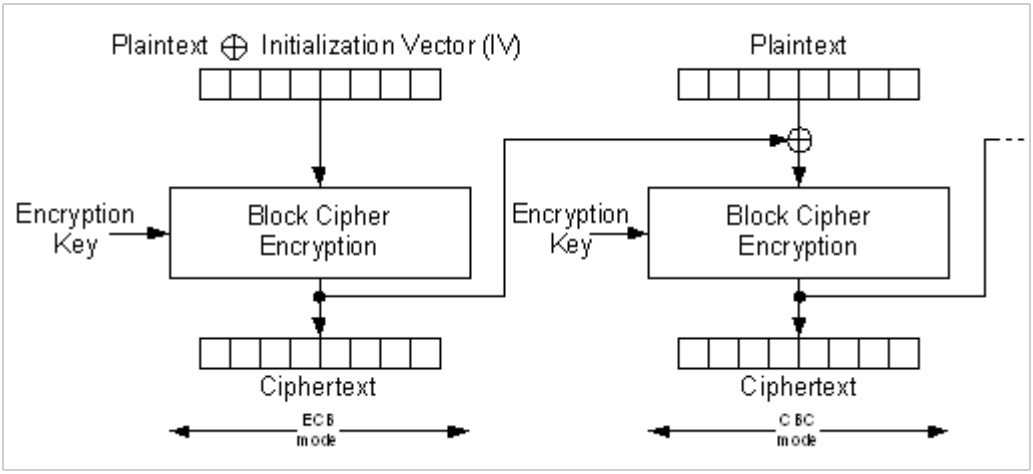
Related Links

14.1.3.3 SRAM Access Mode

14.6.1.4.2 Cipher Block Chaining (CBC)

In CBC mode, the result of a previous AES operation is XORed with the new incoming vector forming the new plaintext to encrypt, see the figure below. This mode is used for the computation of a cryptographic checksum (message integrity code, MIC).

Figure 14-60. CBC Mode – Encryption



After preparing the AES key and defining the AES operation direction using AT86RF212B SRAM register bit AES_DIR, the data has to be provided to the AES engine and the CBC operation can be started.

The first CBC run has to be configured as ECB to process the initial data (plaintext XORed with an initialization vector provided by the microcontroller). All succeeding AES runs are to be configured as CBC by setting the AES_MODE bits in the AES_CTRL register to '2' (AES_CTRL.AES_MODE=2). The AES_DIR bit in the AES_CTRL register (AES_CTRL.AES_DIR) must be set to '0' to enable AES encryption. The data to be processed has to be transferred to the SRAM starting with address 0x84 to 0x93 (AES_STATE registers). Setting the AES_REQUEST bit in the AES_CTRL_MIRROR register to '1' (AES_CTRL_MIRROR.AES_REQUEST=1) starts the first encryption within one SRAM access. This causes the next 128 bits of plaintext data to be XORed with the previous cipher text data.

According to IEEE 802.15.4 the input for the very first CBC operation has to be prepared by a XORing a plaintext with an initialization vector (IV). The value of the initialization vector is zero. However, for non-compliant usage any other initialization vector can be used. This operation has to be prepared by the microcontroller.

Note: The IEEE 802.15.4-2006 standard MIC algorithm requires CBC mode encryption only, as it implements a one-way hash function.

Related Links

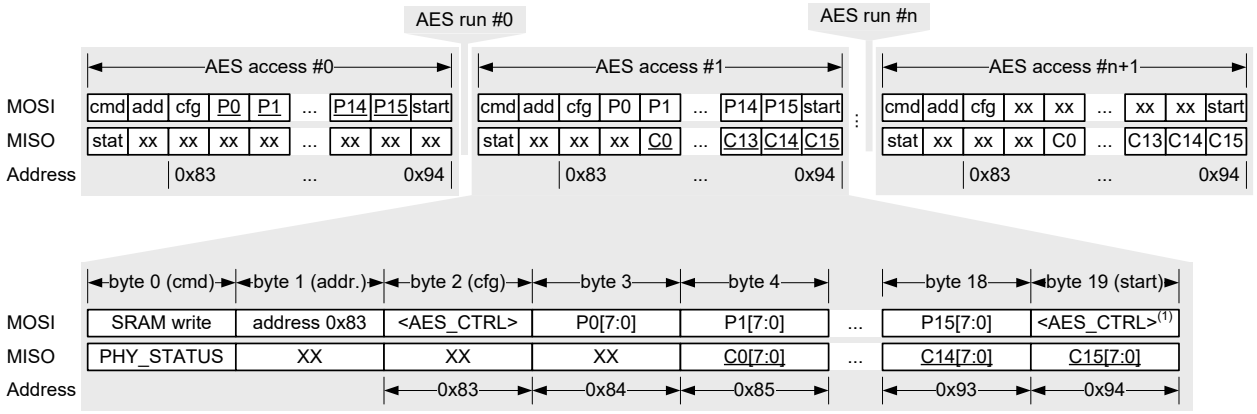
14.6.1.4 Security Operation Modes

14.6.1.5 Data Transfer – Fast SRAM Access

The ECB and CBC modules including the AES core are clocked with 16MHz. One AES operation takes t_{AES} = 23.4µs to execute. That means that the processing of the data is usually faster than the transfer of the data via the SPI interface.

To reduce the overall processing time, the AT86RF212B provides a Fast SRAM access for the address space 0x82 to 0x94.

Figure 14-61. Packet Structure – Fast SRAM Access Mode



Note: Byte 19 is the mirrored version of register AES_CTRL on SRAM address 0x94, see register description AES_CTRL_MIRROR for details.

In contrast to a standard SRAM access, the Fast SRAM access allows writing and reading of data simultaneously during one SPI access for consecutive AES operations (*AES run*).

For each byte P0 transferred to pin 22 (MOSI) for example in "*AES access #1*", see the figure above (lower part), the previous content of the respective AES register C0 is clocked out at pin 20 (MISO) with an offset of one byte.

In the example shown in the above figure the initial plaintext P0 – P15 is written to the SRAM within "*AES access #0*". The last command on address 0x94 (AES_CTRL_MIRROR) starts the AES operation ("*AES run #0*"). In the next "*AES access #1*" new plaintext data P0 – P15 is written to the SRAM for the second AES run, in parallel the ciphertext C0 – C15 from the first AES run is clocked out at pin MISO. To read the ciphertext from the last "*AES run #(n)*" one dummy "*AES access #(n+1)*" is needed.

Note: The SRAM write access always overwrites the previous processing result.

The Fast SRAM access automatically applies to all write operations to SRAM addresses 0x82 to 0x94.

Related Links

14.1.3.3 SRAM Access Mode
14.2.1.4.6 State Transition Timing Summary

14.6.1.6 Start of Security Operation and Status

A security operation is started within one AT86RF212B SRAM access by appending the start command AES_REQUEST = 1 (register 0x94, AES_CTRL_MIRROR) to the SPI sequence. Register AES_CTRL_MIRROR is a mirrored version of register 0x83 (AES_CTRL).

The status of the security processing is indicated by register 0x82 (AES_STATUS). After t_{AES} = 24µs (max.) AES processing time register bit AES_DONE changes to one (register 0x82, AES_STATUS) indicating that the security operation has finished.

14.6.1.7 SRAM Register Summary

The following registers are required to control the security module:

Table 14-35. SRAM Security Module Address Space Overview

SRAM-Addr.	Register Name	Description
0x80 – 0x81		Reserved

.....continued		
SRAM-Addr.	**Register Name**	**Description**
0x82	AES_STATUS	AES status
0x83	AES_CTRL	Security module control, AES mode
0x84 – 0x93	AES_KEY AES_STATE	Depends on AES_MODE setting: AES_MODE = 1: - Contains AES_KEY (key) AES_MODE = 0 or 2: - Contains AES_STATE (128 bit data block)
0x94	AES_CTRL_MIRROR	Mirror of register 0x83 (AES_CTRL)
0x95 – 0xFF		Reserved

These registers are only accessible using SRAM write and read; for details.

Related Links

14.1.3.3 SRAM Access Mode

14.6.1.7.1 AES_CTRL

Name: AES_CTRL
Offset: 0x83
Reset: 0x00
Property: -

The AES_CTRL register controls the operation of the security module.

Notes:

1. Do not access this register during AES operation to read the AES core status. A read or write access during AES operation stops the actual processing.
2. To read the AES status use register bit AES_DONE (register 0x82, AES_STATUS).

Bit	7	6	5	4	3	2	1	0
	AES_REQUEST	AES_MODE[2:0]			AES_DIR			
Access	W	R/W	R/W	R/W	R/W	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – AES_REQUEST AES_REQUEST
A write access with AES_REQUEST = 1 initiates the AES operation.

Table 14-36. AES_REQUEST

Value	Description
0x0	Security module, AES core idle
0x1	A write access starts the AES operation

Bits 6:4 – AES_MODE[2:0] AES_MODE
This register bit sets the AES operation mode.

Table 14-37. AES_MODE

Value	Description
0x0	ECB mode
0x1	KEY mode
0x2	CBC mode
0x3 - 0x7	Reserved

Bit 3 – AES_DIR AES_DIR
The register bit AES_DIR sets the AES operation direction, either encryption or decryption.

Table 14-38. AES_DIR

Value	Description
0x0	AES encryption (ECB, CBC)
0x1	AES decryption (ECB)

14.6.1.7.2 AES_CTRL_MIRROR

Name: AES_CTRL_MIRROR
Offset: 0x94
Reset: 0x00
Property: -

The AES_CTRL_MIRROR register is a mirrored version of the AES_CTRL register.

Notes:

1. Do not access this register during AES operation to read the AES core status. A read or write access during AES operation stops the actual processing.
2. To read the AES status use register bit AES_DONE (register 0x82, AES_STATUS).

Bit	7	6	5	4	3	2	1	0
	AES_REQUEST	AES_MODE[2:0]			AES_DIR			
Access	W	R/W	R/W	R/W	R/W	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – AES_REQUEST AES_REQUEST
A write access with AES_REQUEST = 1 initiates the AES operation.

Table 14-39. AES_REQUEST

Value	Description
0x0	Security module, AES core idle
0x1	A write access starts the AES operation

Bits 6:4 – AES_MODE[2:0] AES_MODE
This register bit sets the AES operation mode.

Table 14-40. AES_MODE

Value	Description
0x0	ECB mode
0x1	KEY mode
0x2	CBC mode
0x3 - 0x7	Reserved

Bit 3 – AES_DIR AES_DIR
The register bit AES_DIR sets the AES operation direction, either encryption or decryption.

Table 14-41. AES_DIR

Value	Description
0x0	AES encryption (ECB, CBC)
0x1	AES decryption (ECB)

14.6.1.7.3 AES_STATUS

Name: AES_STATUS
Offset: 0x82
Reset: 0x00
Property: -

The read-only register AES_STATUS signals the status of the security module and operation.

Bit	7	6	5	4	3	2	1	0
	AES_ER							AES_DONE
Access	R/W	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – AES_ER AES_ER
This SRAM register bit indicates an error of the AES module. An error may occur for instance after an access to SRAM register 0x83 (AES_CTRL) while an AES operation is running or after reading less than 128-bits from SRAM register space 0x84 – 0x93 (AES_STATE).

Table 14-42. AES_ER

Value	Description
0x0	No error of the AES module
0x1	AES module error

Bit 0 – AES_DONE AES_DONE
The bit AES_DONE signals the status of AES operation.

Table 14-43. AES_DONE

Value	Description
0x0	AES operation has not been completed
0x1	AES operation has been completed

14.6.2 Random Number Generator

14.6.2.1 Overview

The AT86RF212B incorporates a two bit truly random number generator by observation of noise. This random number can be used to:

- Generate random seeds for CSMA-CA algorithm
- Generate random values for AES key generation

Random numbers are stored in the RND_VALUE bits in the PHY_RSSI register (PHY_RSSI.RND_VALUE). The random number is updated at every read access in Basic Operating Mode receive states (RX_ON, BUSY_RX). The Random Number Generator does not work if the preamble detector is disabled (RX_SYN.RX_PDT_DIS = 1).

Related Links
14.2.1 Operating Modes
14.2.2 Extended Operationg Mode
14.6.1 Security Module (AES)
14.8.6 PHY_RSSI
14.8.21 RX_SYN

14.6.3 Antenna Diversity

The Antenna Diversity implementation is characterized by:

- Improves signal path robustness between nodes
- Self-contained TX antenna diversity algorithm

- Direct register based antenna selection

14.6.3.1 Overview

Due to multipath propagation effects between network nodes, the receive signal strength may vary and affect the link quality, even for small variance of the antenna location. These fading effects can result in an increased error floor or loss of the connection between devices.

To improve the reliability of an RF connection between network nodes Antenna Diversity can be applied to reduce effects of multipath propagation and fading. Antenna Diversity uses two antennas to select the most reliable RF signal path. To ensure highly independent receive signals on both antennas, the antennas should be carefully separated from each other. The AT86RF212B supports PHY controlled antenna diversity in TX_ARET mode and software controlled antenna diversity (the microcontroller controls which antenna is used for transmission and reception) in Basic and Extended Operating Modes.

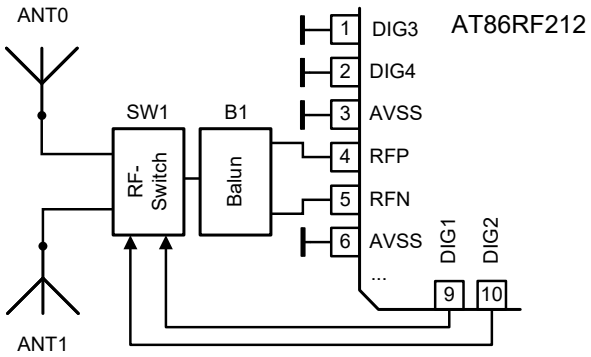
Related Links

14.2.1 Operating Modes
14.2.2 Extended Operationg Mode

14.6.3.2 Antenna Diversity Application Example

A block diagram for an application using an antenna switch is shown in the figure below.

Figure 14-62. Antenna Diversity – Block Diagram



Generally, when the external RF-Switch (SW1) is to be controlled by antenna diversity algorithm, the antenna diversity enable must be activated by the ANT_EXT_SW_EN bit in the ANT_DIV register (ANT_DIV.ANT_EXT_SW_EN). Then the digital control pins pin 9 (DIG1) and pin 10 (DIG2) are enabled to drive the antenna switch control signals to the differential inputs of the RF Switch (SW1) to switch between ANT0 and ANT1.

Related Links

14.8.13 ANT_DIV

14.6.4 RX/TX Indicator

The main features are:

- RX/TX indicator to control an external RF front-end
- Microcontroller independent RF front-end control
- Providing TX timing information

14.6.4.1 Overview

While IEEE 802.15.4 is targeting low cost and low power applications, solutions supporting higher transmit output power are occasionally desirable. To simplify the control of an optional external RF front-end, a differential control pin pair can indicate that the AT86RF212B is currently in transmit mode.

The control of an external RF front-end is done via digital control pins DIG3/DIG4. The function of this pin pair is enabled with the PA_EXT_EN bit in the TRX_CTRL_1 register (TRX_CTRL_1.PA_EXT_EN). While the transmitter is turned off, pin 1 (DIG3) is set to low level and pin 2 (DIG4) to high level. If the radio transceiver starts to transmit, the two pins change the polarity. This differential pin pair can be used to control PA, LNA, and RF switches.

If the AT86RF212B is not in a receive or transmit state, it is recommended to disable the PA_EXT_EN bit to reduce the power consumption or avoid leakage current of external RF switches and other building blocks, especially during SLEEP state. If register bits PA_EXT_EN = 0, output pins DIG3/DIG4 are pulled-down to analog ground.

Related Links

14.8.4 TRX_CTRL_1

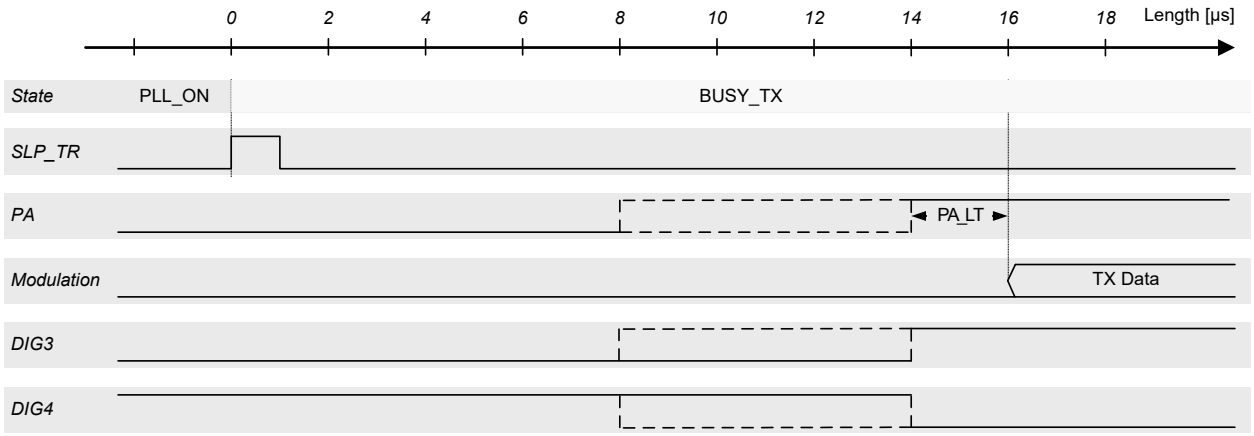
14.6.4.2 External RF Front End Control

When using an external RF front-end including a power amplifier (PA), it may be required to adjust the setup time of the external PA relative to the internal building blocks to optimize the overall power spectral density (PSD) mask.

The start-up sequence of the individual building blocks of the internal transmitter is shown in the figure below where transmission is actually initiated by the rising edge of pin 11 (SLP_TR). The radio transceiver state changes from PLL_ON to BUSY_TX and the PLL settles to the transmit frequency within one symbol period. The modulation starts one symbol period after the rising edge of SLP_TR. During this time, the internal PA is initialized.

The control of the external PA is done via the differential pin pair DIG3/DIG4. DIG3 = H / DIG4 = L indicates that the transmission starts and can be used to enable the external PA. The timing of pins DIG3/DIG4 can be adjusted relative to the start of the frame using the PA_LT bits in the RF_CTRL_0 register (RF_CTRL_0.PA_LT).

Figure 14-63. TX Power Up Ramping Control of RF Front-End for 250kb/s O-QPSK mode



Related Links

14.4.3.5 TX Power Ramping

14.8.22 RF_CTRL_0

14.6.5 RX Frame Time Stamping

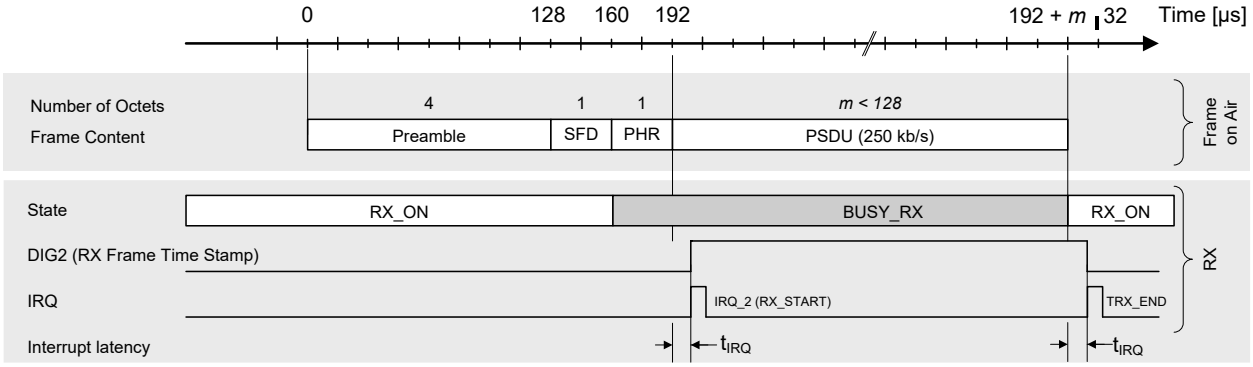
14.6.5.1 Overview

To determine the exact timing of an incoming frame, for example for beaconing networks, the reception of this frame can be signaled to the microcontroller via AT86RF212B pin 10 (DIG2). The pin turns from L to H after detection of a valid PHR. When enabled, DIG2 is set to DIG2 = H at the same time as IRQ_2 (RX_START) occurs, even if IRQ_2 (RX_START) is disabled. The pin remains high for the length of the frame receive procedure.

This function is enabled with the IRQ_2_EXT_EN bit in the TRX_CTRL_1 register (TRX_CTRL_1.IRQ_2_EXT_EN). Pin 10 (DIG2) can be connected to a timer capture unit of the microcontroller.

If this pin is not used for RX Frame Time Stamping, it can be configured for Antenna Diversity. Otherwise, this pin is internally connected to ground.

Figure 14-64. Timing of RX_START and DIG2 for RX Frame Time Stamping within 250kb/s O-QPSK mode



1. Timing figures t_{IRQ} refer to the *Digital Interface Timing Characteristics*.

Related Links

14.6.3 Antenna Diversity
14.8.4 TRX_CTRL_1

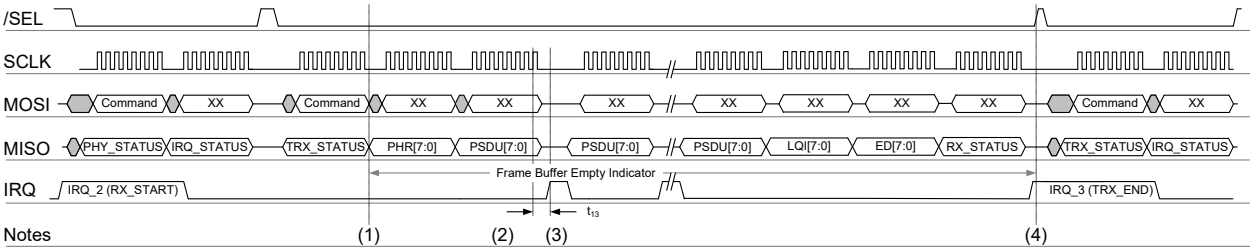
14.6.6 Frame Buffer Empty Indicator

14.6.6.1 Overview

For time critical applications that want to start reading the frame data as early as possible, the Frame Buffer status can be indicated to the microcontroller through a dedicated pin. This pin indicates to the microcontroller if an access to the Frame Buffer is not possible since valid PSDU data are missing.

Pin 24 (IRQ) can be configured as a Frame Buffer Empty Indicator during a Frame Buffer read access. This mode is enabled by the RX_BL_CTRL bit in the TRX_CTRL_1 register (TRX_CTRL_1RX_BL_CTRL). The IRQ pin turns into Frame Buffer Empty Indicator after the Frame Buffer read access command, see note *(1)* below, has been transferred on the SPI bus until the Frame Buffer read procedure has finished indicated by /SEL = H, see note *(4)*.

Figure 14-65. Timing Diagram of Frame Buffer Empty Indicator



Notes:

1. Timing figure t_{12} refer to the *Digital Interface Timing Characteristics*.
2. A Frame Buffer read access can proceed as long as pin 24 (IRQ) = L.
3. Pin IRQ = H indicates that the Frame Buffer is currently not ready for another SPI cycle.
4. The Frame Buffer read procedure has finished indicated by /SEL = H.

The microcontroller has to observe the IRQ pin during the Frame Buffer read procedure. A Frame Buffer read access can proceed as long as pin 24 (IRQ) = L, see note *(2)*. When the IRQ output pin is pulled high (IRQ = H) , the Frame Buffer is not ready for another SPI cycle, see note *(3)* above. The read operation can be resumed as the IRQ output pin is pulled low again (IRQ = L) to indicate new data in the buffer.

On Frame Buffer read access, three more byte are transferred via MISO after PHR and PSDU data, namely LQI, ED, and RX_STATUS. Because these bytes are appended and physically not stored in the frame buffer, they are ignored for Frame Buffer empty indication.

The Frame Buffer Empty Indicator pin 24 (IRQ) becomes valid after t_{12} = 750ns starting from the last SCLK rising edge while reading a Frame Buffer command byte, see figure above.

Upon completing the SPI frame data receive task, SPI read access can be disabled by pulling /SEL = H, note *(4)*. At this time the IRQ output pin 24 (IRQ)) can be used as an output to flag pending interrupts to the processor.

If during the Frame Buffer read access a receive error occurs (for example an PLL unlock), the Frame Buffer Empty Indicator locks on 'empty' (pin 24 (IRQ) = H) too. To prevent possible deadlocks, the microcontroller should impose a timeout counter that checks whether the Frame Buffer Empty Indicator remains logic high for more than two octet periods. A new byte must have been arrived at the frame buffer during that period. If not, the Frame Buffer read access should be aborted.

Related Links

14.1.3.2 Frame Buffer Access Mode

14.8.4 TRX_CTRL_1

14.6.7 Dynamic Frame Buffer Protection

14.6.7.1 Overview

The AT86RF212B continues the reception of incoming frames as long as it is in any receive state. When a frame was successfully received and stored into the Frame Buffer, the following frame will overwrite the Frame Buffer content again.

To relax the timing requirements for a Frame Buffer read access the Dynamic Frame Buffer Protection prevents that a new valid frame passes to the Frame Buffer until a Frame Buffer read access has ended (indicated by /SEL = H).

A received frame is automatically protected against overwriting:

- in Basic Operating Mode, if its FCS is valid.
- in Extended Operating Mode, if an IRQ_3 (TRX_END) is generated.

The Dynamic Frame Buffer Protection is enabled with the RX_SAFE_MODE bit in the TRX_CTRL_2 register (TRX_CTRL_2.RX_SAFE_MODE) set and applicable in transceiver states RX_ON and RX_AACK_ON.

Note: The Dynamic Frame Buffer Protection only prevents write accesses from the air interface – not from the SPI interface. A Frame Buffer or SRAM write access may still modify the Frame Buffer content.

Related Links

14.1.3 SPI Protocol

14.2.1 Operating Modes

14.2.2 Extended Operationg Mode

14.8.12 TRX_CTRL_2

14.6.8 Alternate Start-Of-Frame Delimiter

14.6.8.1 Overview

The Start of Frame Delimiter (SFD) is a field indicating the end of the SHR and the start of the packet data. The length of the SFD is one octet (eight symbols for BPSK and two symbols for O-QPSK). The octet is used for byte synchronization only and is not included in the AT86RF212B Frame Buffer.

The value of the SFD can be changed if it is needed to operate in non-IEEE 802.15.4 compliant networks. A node with a non-standard SFD value cannot synchronize with any of the IEEE 802.15.4 network nodes.

Due to the way the SHR is formed, it is not recommended to set the low-order four bits to zero. The LSB of the SFD is transmitted first, that is right after the last bit of the preamble sequence.

Related Links

14.8.11 SFD_VALUE

14.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	Reserved									
0x01	TRX_STATUS	7:0	CCA_DONE	CCA_STATUS		TRX_STATUS[4:0]				
0x02	TRX_STATE	7:0	TRAC_STATUS[2:0]			TRX_CMD[4:0]				
0x03	TRX_CTRL_0	7:0	PAD_IO[1:0]		PAD_IO_CLKM[1:0]		CLKM_SHA_SEL		CLKM_CTRL[1:0]	
0x04	TRX_CTRL_1	7:0	PA_EXT_EN	IRQ_2_EXT_EN	TX_AUTO_CRC_ON	RX_BL_CTRL	SPI_CMD_MODE[1:0]		IRQ_MASK_MODE	IRQ_POLARITY
0x05	PHY_TX_PWR	7:0	PA_BOOST	GC_PA[1:0]		TX_PWR[4:0]				
0x06	PHY_RSSI	7:0	RX_CRC_VALID	RND_VALUE[1:0]		RSSI[4:0]				
0x07	PHY_ED_LEVEL	7:0	ED_LEVEL[7:0]							
0x08	PHY_CC_CCA	7:0	CCA_REQUEST	CCA_MODE[1:0]		CHANNEL[4:0]				
0x09	CCA_THRES	7:0					CCA_CS_THRES[3:0]			
0x0A	RX_CTRL	7:0			JCM_EN					
0x0B	SFD_VALUE	7:0	SFD_VALUE[7:0]							
0x0C	TRX_CTRL_2	7:0	RX_SAFE_MODE	TRX_OFF_AVDD_EN	OQPSK_SCRAM_EN	ALT_SPECTRUM	BPSK_OQPSK	SUB_MODE	OQPSK_DATA_RATE[1:0]	
0x0D	ANT_DIV	7:0	ANT_SEL				ANT_DIV_EN	ANT_EXT_SW_EN	ANT_CTRL[1:0]	
0x0E	IRQ_MASK	7:0	IRQ_MASK[7:0]							
0x0F	IRQ_STATUS	7:0	IRQ_7_BAT_LOW	IRQ_6_TRX_UR	IRQ_5_AMI	IRQ_4_CCA_ED_DONE	IRQ_3_TRX_END	IRQ_2_RX_START	IRQ_1_PLL_UNLOCK	IRQ_0_PLL_LOCK
0x10	VREG_CTRL	7:0	AVREG_EXT	AVDD_OK			DVREG_EXT	DVDD_OK		
0x11	BATMON	7:0	PLL_LOCK_CP		BATMON_OK	BATMON_HR	BATMON_VTH[3:0]			
0x12	XOSC_CTRL	7:0	XTAL_MODE[3:0]				XTAL_TRIM[3:0]			
0x13	CC_CTRL_0	7:0	CC_NUMBER[7:0]							
0x14	CC_CTRL_1	7:0						CC_BAND[2:0]		
0x15	RX_SYN	7:0	RX_PDT_DIS	RX_OVERRIDE[2:0]			RX_PDT_LEVEL[3:0]			
0x16	RF_CTRL_0	7:0	PA_LT[1:0]						GC_TX_OFFS[1:0]	
0x17	XAH_CTRL_1	7:0		CSMA_LBT_MODE	AACK_FLTR_RES_FT	AACK_UPLD_RES_FT		AACK_ACK_TIME	AACK_PROM_MODE	
0x18	FTN_CTRL	7:0	FTN_START							
0x19	Reserved									
0x1A	PLL_CF	7:0	PLL_CF_START							
0x1B	PLL_DCU	7:0	PLL_DCU_START							
0x1C	PART_NUM	7:0	PART_NUM[7:0]							
0x1D	VERSION_NUM	7:0	VERSION_NUM[7:0]							
0x1E	MAN_ID_0	7:0	MAN_ID_0[7:0]							
0x1F	MAN_ID_1	7:0	MAN_ID_1[7:0]							
0x20	SHORT_ADDR_0	7:0	SHORT_ADDR_0[7:0]							
0x21	SHORT_ADDR_1	7:0	SHORT_ADDR_1[15:8]							
0x22	PAN_ID_0	7:0	PAN_ID_0[7:0]							
0x23	PAN_ID_1	7:0	PAN_ID_1[7:0]							
0x24	IEEE_ADDR_0	7:0	IEEE_ADDR[7:0]							
0x25	IEEE_ADDR_1	7:0	IEEE_ADDR[7:0]							
0x26	IEEE_ADDR_2	7:0	IEEE_ADDR[7:0]							
0x27	IEEE_ADDR_3	7:0	IEEE_ADDR[7:0]							
0x28	IEEE_ADDR_4	7:0	IEEE_ADDR[7:0]							
0x29	IEEE_ADDR_5	7:0	IEEE_ADDR[7:0]							
0x2A	IEEE_ADDR_6	7:0	IEEE_ADDR[7:0]							
0x2B	IEEE_ADDR_7	7:0	IEEE_ADDR[7:0]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	XAH_CTRL_0	7:0	MAX_FRAME_RETRIES[3:0]				MAX_CSMA_RETRIES[2:0]			SLOTTED_OPERATION
0x2D	CSMA_SEED_0	7:0	CSMA_SEED_0[7:0]							
0x2E	CSMA_SEED_1	7:0	AACK_FVN_MODE[1:0]		AACK_SET_PD	AACK_DIS_ACK	AACK_I_AM_COORD	CSMA_SEED_1[2:0]		
0x2F	CSMA_BE	7:0	MAX_BE[3:0]				MIN_BE[3:0]			

14.8 Register Description

14.8.1 TRX_STATUS

Name: TRX_STATUS
Offset: 0x01
Reset: 0x00
Property: -

The read-only register TRX_STATUS signals the present state of the radio transceiver as well as the status of a CCA operation.

Bit	7	6	5	4	3	2	1	0
	CCA_DONE	CCA_STATUS		TRX_STATUS[4:0]				
Access	R	R		R	R	R	R	R
Reset	0	0		0	0	0	0	0

Bit 7 – CCA_DONE CCA_DONE
The CCA_DONE bit indicates if a CCA request is completed. This is also indicated by an interrupt IRQ_4 (CCA_ED_DONE). The register bit CCA_DONE is cleared in response to a CCA_REQUEST.

Value	Description
0x0	CCA calculation not finished
0x1	CCA calculation finished

Bit 6 – CCA_STATUS CCA_STATUS
After a CCA request is completed, the result of the CCA measurement is available in the CCA_STATUS bit. The register bit CCA_STATUS is cleared in response to a CCA_REQUEST.

Value	Description
0x0	Channel indicated as busy
0x1	Channel indicated as idle

Bits 4:0 – TRX_STATUS[4:0] TRX_STATUS
The register bits TRX_STATUS signal the current radio transceiver status.

Table 14-44. TRX_STATUS

Value	Description
0x00	P_ON
0x01	BUSY_RX
0x02	BUSY_TX
0x06	RX_ON
0x08	TRX_OFF (CLK Mode)
0x09	PLL_ON (TX_ON)
0x0F(1)	SLEEP
0x11(2)	BUSY_RX_AACK
0x12(2)	BUSY_TX_ARET
0x16(2)	RX_AACK_ON
0x19(2)	TX_ARET_ON
0x1C	RX_ON_NOCLK
0x1D	RX_AACK_ON_NOCLK
0x1E	BUSY_RX_AACK_NOCLK
0x1F(3)	STATE_TRANSITION_IN_PROGRESS
Reserved	All other values are reserved

1. In SLEEP or DEEP_SLEEP state register not accessible.
2. Extended Operating Mode only.

3. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS state.

A read access to register bits TRX_STATUS reflects the current radio transceiver state. A state change is initiated by writing a state transition command to register bits TRX_CMD (register 0x02, TRX_STATE). Alternatively, some state transitions can be initiated by the rising edge of SLP_TR in the appropriate state.
These register bits are used for Basic and Extended Operating Mode.
If the requested state transition has not been completed, the TRX_STATUS returns STATE_TRANSITION_IN_PROGRESS value. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS state.

14.8.2 TRX_STATE

Name: TRX_STATE
Offset: 0x02
Reset: 0x00
Property: -

The radio transceiver states are advanced via register TRX_STATE by writing a command word into register bits TRX_CMD. The read-only register bits TRAC_STATUS indicate the status or result of an Extended Operating Mode transaction.

Bit	7	6	5	4	3	2	1	0
	TRAC_STATUS[2:0]			TRX_CMD[4:0]				
Access	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – TRAC_STATUS[2:0] TRAC_Status
The status of the RX_AACK and TX_ARET procedure is indicated by register bits TRAC_STATUS. Values are meaningful after an interrupt until the next frame transmit. Details of the algorithm and a description of the status information are given in “RX_AACK_ON – Receive with Automatic ACK” on page 919 and “TX_ARET_ON – Transmit with Automatic Frame Retransmission and CSMA-CA Retry” on page 929.

Value	Description	RX_AACK	TX_ARET
0x0(1)	SUCCESS	X	X
0x1	SUCCESS_DATA_PENDING		X
0x2	SUCCESS_WAIT_FOR_ACK	X	
0x3	CHANNEL_ACCESS_FAILURE		X
0x5	NO_ACK		X
0x7(1)	INVALID	X	X
-	All other values are reserved		

1. Even though the reset value for register bits TRAC_STATUS is zero, the RX_AACK and TX_ARET procedures set the register bits to TRAC_STATUS = 7 (INVALID) when they are started.

The status of the RX_AACK and TX_ARET procedures is indicated by register bits TRAC_STATUS. Values are meaningful after an interrupt until the next frame transmit. Details of the algorithms and a description of the status information are given in the RX_AACK_ON and TX_ARET_ON sections.

RX_AACK
SUCCESS_WAIT_FOR_ACK: Indicates an ACK frame is about to be sent in RX_AACK slotted acknowledgement. Slotted acknowledgement operation must be enabled with register bit SLOTTED_OPERATION (register 0x2C, XAH_XTRL_0). The microcontroller must pulse pin 11 (SLP_TR) at the next backoff slot boundary in order to initiate a transmission of the ACK frame. For details refer to IEEE 802.15.4-2006, Section 7.5.6.4.2.

TX_ARET
SUCCESS_DATA_PENDING: Indicates a successful reception of an ACK frame with frame pending bit set to one.

Bits 4:0 – TRX_CMD[4:0] TRX_CMD
A write access to register bits TRX_CMD initiate a radio transceiver state transition to the new state.

Table 14-45. TRX_CMD

Value	Description
0x00(1)	NOP
0x02(2)	TX_START
0x03	FORCE_TRX_OFF
0x04(3)	FORCE_PLL_ON
0x06	RX_ON
0x08	TRX_OFF (CLK Mode)
0x09	PLL_ON (TX_ON)

.....continued	
Value	**Description**
0x16[4]	RX_AACK_ON
0x19[4]	TX_ARET_ON
Reserved	All other values are reserved

1. TRX_CMD = 0 after power on reset (POR).
2. The frame transmission starts one symbol after TX_START command.
3. FORCE_PLL_ON is not valid for states P_ON, SLEEP, RESET and all *_NOCLK states, as well as STATE_TRANSITION_IN_PROGRESS towards these states.
4. Extended Operating Mode only.

A write access to register bits TRX_CMD initiates a radio transceiver state transition towards the new state. These register bits are used for Basic and Extended Operating Mode, see “Extended Operating Mode” on page 916

14.8.3 TRX_CTRL_0

Name: TRX_CTRL_0
Offset: 0x03
Reset: 0x19
Property: -

The TRX_CTRL_0 register controls the driver current of the digital output pads and the CLKM clock rate.

Bit	7	6	5	4	3	2	1	0
	PAD_IO[1:0]		PAD_IO_CLKM[1:0]		CLKM_SHA_SEL		CLKM_CTRL[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	1	1		0	1

Bits 7:6 – PAD_IO[1:0] PAD_IO
These register bits set the output driver current of digital output pads, except CLKM.

Value	Description
0x0	2mA
0x1	4mA
0x2	6mA
0x3	8mA

1. Selecting low-level driver current reduces power consumption and minimizes transceiver's harmonic distorion.

Bits 5:4 – PAD_IO_CLKM[1:0] PAD_IO_CLKM
These register bits set the output driver current of pin CLKM. It is recommended to reduce the driver strength to 2mA (PAD_IO_CLKM = 0) if possible. This reduces power consumption and spurious emissions.

Value	Description
0x0	2mA
0x1	4mA
0x2	6mA
0x3	8mA

Bit 3 – CLKM_SHA_SEL CLKM_SHA_SEL
The register bit CLKM_SHA_SEL defines whether a new clock rate (defined by CLKM_CTRL) is set immediately or gets effective after the next SLEEP cycle.

Value	Description
0x0	CLKM clock rate change appears immediately
0x1	CLKM clock rate change appears after SLEEP cycle

Bits 1:0 – CLKM_CTRL[1:0] CLKM_CTRL
The register bits CLKM_CTRL set the clock rate of CLKM.

Table 14-46. CLKM_CTRL

Value	Description
0x0	No clock at CLKM, signal set to logic low
0x1	1MHz
0x2	2MHz
0x3	4MHz
0x4	8MHz
0x5	16MHz
0x6	250kHz
0x7	62.5kHz (IEEE 802.15.4 symbol rate)

1. If a clock rate is selected between 1MHz and 16MHz and SLP_TR is set to logic high in state TRX_OFF, the TRX delivers additional 35 clock cycles before entering state SLEEP.

14.8.4 TRX_CTRL_1

Name: TRX_CTRL_1
Offset: 0x04
Reset: 0x22
Property: -

The TRX_CTRL_1 register is a multi-purpose register to control various operating modes and settings of the radio transceiver.

Bit	7	6	5	4	3	2	1	0
	PA_EXT_EN	IRQ_2_EXT_EN	TX_AUTO_CRC_ON	RX_BL_CTRL	SPI_CMD_MODE[1:0]		IRQ_MASK_MODE	IRQ_POLARITY
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	1	0

Bit 7 – PA_EXT_EN PA_EXT_EN
The register bit PA_EXT_EN enables RF front-end control signals DIG3 and DIG4 to indicate the transmit state of the radio transceiver.

Table 14-47. PA_EXT_EN

PA_EXT_EN	State	Signal	Value	Description
0x0	N/A	DIG3	L	External RF front-end control disabled
		DIG4	L	
0x1(1)	TX_BUSY	DIG3	H	External RF front-end control enabled
		DIG4	L	
	Other	DIG3	L	
		DIG4	H	

1. It is recommended to set PA_EXT_EN = 1 only in receive or transmit states to reduce the power consumption or avoid leakage current of external RF switches or other building blocks, especially during SLEEP or DEEP_SLEEP state.

Bit 6 – IRQ_2_EXT_EN IRQ_2_EXT_EN
The register bit IRQ_2_EXT_EN controls external signaling for time stamping via DIG2.

Table 14-48. IRQ_2_EXT_EN

Value	Description
0x0	Time stamping over pin 10 (DIG2) is disabled
0x1(1)	Time stamping over pin 10 (DIG2) is enabled

1. The pin 10 (DIG2) is also active if the corresponding interrupt event IRQ_2 (RX_START) mask bit in register 0x0E (IRQ_MASK) is set to zero.

The timing of a received frame can be determined by a separate pin 10 (DIG2). If register bit IRQ_2_EXT_EN is set to one, the reception of a PHR field is directly issued on pin 10 (DIG2), similar to interrupt IRQ_2 (RX_START).

Bit 5 – TX_AUTO_CRC_ON TX_AUTO_CRC_ON
The register bit TX_AUTO_CRC_ON controls the automatic FCS generation for transmit operations.

Table 14-49. TX_AUTO_CRC_ON

Value	Description
0x0	Automatic FCS generation is disabled
0x1	Automatic FCS generation is enabled

1. The TX_AUTO_CRC_ON function can be used within Basic and Extended Operating Modes.

Bit 4 – RX_BL_CTRL RX_BL_CTRL
The register bit RX_BL_CTRL controls the Frame Buffer Empty Indicator.

Table 14-50. RX_BL_CTRL

Value	Description
0x0	Frame Buffer Empty Indicator disabled
0x1	Frame Buffer Empty Indicator enabled

1. A modification of register bit IRQ_POLARITY has no influence to RX_BL_CTRL behavior.

If this register bit is set, the Frame Buffer Empty Indicator is enabled. After sending a Frame Buffer read command, signal IRQ indicates that an access to the Frame Buffer is not possible since PSDU data are not available yet. The IRQ signal does not indicate any interrupts during this time.

Bits 3:2 – SPI_CMD_MODE[1:0] SPI_CMD_MODE
Each SPI transfer returns bytes back to the SPI master. The content of the first byte (PHY_STATUS) can be configured using register bits SPI_CMD_MODE.

Table 14-51. SPI_CMD_MODE

Value	Description
0x0	No clock at CLKM, signal set to logic low
0x1	Monitor TRX_STATUS register
0x2	Monitor PHY_RSSI register
0x3	Monitor IRQ_STATUS register

Bit 1 – IRQ_MASK_MODE IRQ_MASK_MODE
The radio transceiver supports polling of interrupt events. Interrupt polling is enabled by setting register bit IRQ_MASK_MODE.

Table 14-52. IRQ_MASK_MODE

Value	Description
0x0	Interrupt polling is disabled. Masked off IRQ bits will not appear in IRQ_STATUS register.
0x1	Interrupt polling is enabled. Masked off IRQ bits will appear in IRQ_STATUS register.

With the interrupt polling enabled (IRQ_MASK_MODE = 1) the interrupt events are flagged in the register 0x0F (IRQ_STATUS) when their respective mask bits are disabled in the register 0x0E (IRQ_MASK).

Bit 0 – IRQ_POLARITY IRQ_POLARITY
The register bit IRQ_POLARITY controls the polarity for pin 24 (IRQ). The default polarity of the pin 24 (IRQ) is active high. The polarity can be configured to active low via register bit IRQ_POLARITY.

Table 14-53. IRQ_POLARITY

Value	Description
0x0	IRQ signal is high active.
0x1	IRQ signal is low active.

1. A modification of register bit IRQ_POLARITY has no influence to RX_BL_CTRL behavior.

This setting does not affect the polarity of the Frame Buffer Empty Indicator. The Frame Buffer Empty Indicator is always active high.

14.8.5 PHY_TX_PWR

Name: PHY_TX_PWR
Offset: 0x05
Reset: 0x60
Property: -

The PHY_TX_PWR register controls the output power of the transmitter.

Bit	7	6	5	4	3	2	1	0
	PA_BOOST	GC_PA[1:0]		TX_PWR[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	0	0	0

Bit 7 – PA_BOOST PA_BOOST
The register bit PA_BOOST increases transmit gain by 5dB.

Value	Description
0x0	PA boost mode is disabled
0x1	PA boost mode is enabled

This register bit enables the PA boost mode where the TX output power is increased by approximately 5dB when PA_BOOST = 1. In PA boost mode, the PA linearity is decreased compared to the normal mode when PA_BOOST = 0. This leads to higher spectral side lobes of the TX power spectrum and higher power of the harmonics. Consequently, the higher TX power settings do not fulfill the regulatory requirements of the European 868.3MHz band regarding spurious emissions in adjacent frequency bands (see ETSI EN 300 220-1, ERC/REC 70-03, and ERC/DEC/(01)04).

Bits 6:5 – GC_PA[1:0] GC_PA
The register bits GC_PA control the PA gain.

Value	Description
0x0	-2.9dB
0x1	-1.3dB
0x2	-0.9dB
0x3	0dB

These register bits control the gain of the PA by changing its bias current. GC_PA needs to be set in TRX_OFF mode only. It can be used to reduce the supply current in TX mode when a reduced TX power is selected with the TX_PWR control word. A reduced PA bias current causes lower RF gain and lowers the 1dB compression point of the PA. Hence, it is advisable to use a reduced bias current of the PA only in combination with lower values of TX_PWR. A reasonable combination of register bits TX_PWR and GC_PA is shown in Table 14-54.

Bits 4:0 – TX_PWR[4:0] TX_PWR
The register bits TX_PWR determine the TX output power of the radio transceiver.
These register bits control the transmitter output power measured at pins RFP/RFN. The value of TX_PWR describes the power reduction relative to the maximum output power. The resolution is 1dB per step. Since TX_PWR adjusts the gain in the TX path prior to the PA, the PA bias setting is not optimal for increased values of TX_PWR regarding PA efficiency.
The PA power efficiency can be improved when PA bias is reduced (decreased GC_PA value) along with the TX power setting (increased TX_PWR value). A recommended combination of TX power control (TX_PWR), PA bias control (GC_PA), and PA boost mode (PA_BOOST) is listed in Table 9-15. It is a recommended mapping of intended TX power to the 8-bit word in register 0x05 (PHY_TX_PWR).

Table 14-54. Recommended Mapping of TX Power, Frequency Band, and PHY_TX_PWR (register 0x05).

		PHY_TX_PWR (register 0x05)	
TX Power [dBm]	**915MHz North American Band** **PHY Modes: BPSK-40 (GC_TX_OFFS=3), BPSK-40-ALT (GC_TX_OFFS=3), OQPSK-SIN-{250,500,1000} (GC_TX_OFFS=2)**	**868.3MHz European Band** **PHY Modes: BPSK-20 (GC_TX_OFFS=3), OQPSK-SIN-RC-{100,200,400} (GC_TX_OFFS=2) OQPSK-RC-{100,200,400} (GC_TX_OFFS=3)**	**780MHz Chinese Band** **PHY Modes: OQPSK-RC-{250,500,1000} (GC_TX_OFFS=2)**
11	0xC0	0xA0	0xC1
10	0xC1	0x80	0xE3
9	0x80	0xE4	0xE4
8	0x82	0xE6	0xC5
7	0x83	0xE7	0xE7
6	0x84	0xE8	0xE8
5	0x40	0xE9	0xE9
4	0x86	0xEA	0xEA
3	0x00	0xCB	0xCB
2	0x01	0xCC	0xCC
1	0x02	0xCC	0xCD
0	0x03	0xAD	0xCE
-1	0x04	0x47	0xCF
-2	0x27	0x48	0xAF
-3	0x05	0x49	0x26
-4	0x07	0x29	0x27
-5	0x08	0x90	0x28
-6	0x91	0x91	0x29
-7	0x09	0x93	0x07
-8	0x0B	0x94	0x08
-9	0x0C	0x2F	0x09
-10	0x0D	0x30	0x0A
-11	0x0E	0x31	0x0B
-12	0x0F	0x0F	0x0C
-13	0x10	0x10	0x0D
-14	0x11	0x11	0x0E
-15	0x12	0x12	0x0F
-16	0x13	0x13	0x10
-17	0x14	0x14	0x11
-18	0x15	0x15	0x13
-19	0x16	0x17	0x14
-20	0x17	0x18	0x15
-21	0x19	0x19	0x16
-22	0x1A	0x1A	0x17
-23	0x1B	0x1B	0x18
-24	0x1C	0x1C	0x19
-25	0x1D	0x1D	0x1A

1. Spectral side lobes remain < -54dBm / 100kHz measured with an RMS detector outside F_C ± 3MHz. Power settings may be used at channel 1 and 2 according to IEEE802.15.4c.
2. Spectral side lobes remain < -54dBm / 100kHz measured with an RMS detector outside F_C ± 1MHz. Power settings may be used at channel 0 and 3 according to IEEE802.15.4c.

Values of the above table are based on a mode dependent setting of register bits GC_TX_OFFS (register 0x16, RF_CTRL_0).

14.8.6 PHY_RSSI

Name: PHY_RSSI
Offset: 0x06
Reset: 0x00
Property: -

The PHY_RSSI register is a multi-purpose register that indicates FCS validity, to provide random numbers, and a RSSI value.

Bit	7	6	5	4	3	2	1	0
	RX_CRC_VALID	RND_VALUE[1:0]		RSSI[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RX_CRC_VALID RX_CRC_VALID
The register bit RX_CRC_VALID signals the FCS check status for a received frame.

Table 14-55. RX_CRC_VALID

Value	Description
0x0	FCS is not valid
0x1	FCS is valid

Reading this register bit indicates whether the last received frame has a valid FCS or not. The register bit is updated when issuing interrupt IRQ_3 (TRX_END) and remains valid until the next TRX_END interrupt is issued, caused by a new frame reception.

Bits 6:5 – RND_VALUE[1:0] RND_VALUE
The 2-bit random value can be retrieved by reading register bits RND_VALUE.

Table 14-56. RND_VALUE

Value	Description
0x0	Deliver two bit noise value within receive state. Valid values are [3, 2, …, 0].

1. The radio transceiver shall be in Basic Operating Mode receive state.

Bits 4:0 – RSSI[4:0] RSSI
Received signal strength as a linear curve on a logarithmic input power scale with a resolution of 3.1dB.

Table 14-57. RSSI

Value	Description
0x00	Minimum RSSI value
0x1C	Maximum RSSI value

The result of the automated RSSI measurement is stored in the RSSI bits in the PHY_RSSI register. The value is updated every t_{RSSI} = 2µs in any receive state.
The result of the automated RSSI measurement is stored in register bits RSSI (register 0x06, PHY_RSSI). The value is updated at time intervals according to the *RSSI Update Interval* table in any receive state. RSSI is a number between zero and 28, representing the received signal strength.

14.8.7 PHY_ED_LEVEL

Name: PHY_ED_LEVEL
Offset: 0x07
Reset: 0xFF
Property: -

The PHY_ED_LEVEL register contains the result of an ED measurement.

Bit	7	6	5	4	3	2	1	0
	ED_LEVEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – ED_LEVEL[7:0] ED_LEVEL
The register bits ED_LEVEL signals the ED level for the current channel.

Table 14-58. ED_LEVEL

Value	Description
0x00	Minimum ED level value
0x534	Maximum ED level value
0xFF	Reset value

The measured ED value has a valid range from 0x00 to 0x54 (zero to 84). The value 0xFF signals that no measurement has been started yet (reset value).
A manual ED measurement can be initiated by a write access to the register.

14.8.8 PHY_CC_CCA

Name: PHY_CC_CCA
Offset: 0x08
Reset: 0x25
Property: -

The PHY_CC_CCA register is a multi-purpose register that controls CCA configuration, CCA measurement, and the IEEE 802.15.4 channel setting.

Bit	7	6	5	4	3	2	1	0
	CCA_REQUEST	CCA_MODE[1:0]		CHANNEL[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	0	1

Bit 7 – CCA_REQUEST CCA_REQUEST
The register bit CCA_REQUEST initiates a manual started CCA measurement.

Table 14-59. CCA_REQUEST

Value	Description
0x0	Reset value
0x1	Starts a CCA measurement

1. The read value returns always with zero.
2. If a CCA request is initiated in states others than RX_ON or RX_BUSY the PHY generates an IRQ_4 (CCA_ED_DONE) and sets the register bit CCA_DONE, however no CCA was carried out.

A manual CCA measurement is initiated with setting CCA_REQUEST = 1. The end of the CCA measurement is indicated by interrupt IRQ_4 (CCA_ED_DONE). Register bits CCA_DONE and CCA_STATUS (register 0x01, TRX_STATUS) are updated after a CCA_REQUEST. The register bit is automatically cleared after requesting a CCA measurement with CCA_REQUEST = 1.

Bits 6:5 – CCA_MODE[1:0] CCA_MODE
The CCA mode can be selected using register bits CCA_MODE.

Table 14-60. CCA_MODE

Value	Description
0x0	Mode 3a, Carrier sense OR energy above threshold
0x1	Mode 1, Energy above threshold
0x2	Mode 2, Carrier sense only
0x3	Mode 3b, Carrier sense AND energy above threshold

1. IEEE 802.15.4–2006 CCA mode 3 defines the logical combination of CCA mode 1 and 2 with the logical operators AND or OR.
2. If register bit CSMA_LBT_MODE is set, CCA_MODE configures the LBT measurement.

Bits 4:0 – CHANNEL[4:0] CHANNEL
The register bits CHANNEL define the RX/TX channel. The channel assignment is according to IEEE 802.15.4.

Table 14-61. CHANNEL

Value	Description
0x00	868.3MHz
0x01	906MHz
0x02	908MHz
0x03	910MHz
0x04	912MHz

.....continued	
Value	**Description**
0x05	914MHz
0x06	916MHz
0x07	918MHz
0x08	920MHz
0x09	922MHz
0x0A	924MHz
Reserved	All other values are reserved

14.8.9 CCA_THRES

Name: CCA_THRES
Offset: 0x09
Reset: 0x77
Property: -

The CCA_THRES register sets the ED threshold level for CCA.

Bit	7	6	5	4	3	2	1	0
					CCA_CS_THRES[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	1	1

Bits 3:0 – CCA_CS_THRES[3:0] CCA_CS_THRES
The register bits CCA_CS_THRES are used for CCA carrier sense algorithm.

Table 14-62. CCA_CS_THRES

Value	Description
0x07	Default value.
0x0F	A threshold of 15 always signals an empty channel
	All other values are reserved

1. A value of seven (reset value) corresponds to normal CCA_CS operation. A value of 15 results in always sensing an empty channel if CCA_MODE = 2 (carrier sense only). This can be useful in combination with TX_ARET, that is allowing retries with actually performing CSMA-CA.

Bits 3:0 – CCA_ED_THRES[3:0] CCA_ED_THRES
An ED value above the threshold signals the channel as busy during a CCA_ED measurement.

Table 14-63. CCA_ED_THRES

Value	Description
0x07	For CCA_MODE = 1, a busy channel is indicated if the measured received power is above P_THRES[dBm] = RSSI_BASE_VAL[dBm] + 2[dB] x CCA_ED_THRES. CCA modes 0 and 3 are logically related to this result.

1. If CSMA_LBT_MODE is enabled, CCA_ED_THRES is used for the LBT measurement.

14.8.10 RX_CTRL

Name: RX_CTRL
Offset: 0x0A
Reset: 0x17
Property: -

The register RX_CTRL configures the clock jitter module.

Bit	7	6	5	4	3	2	1	0
			JCM_EN					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	1	1	1

Bit 5 – JCM_EN JCM_EN
The register bit JCM_EN controls digital clock jitter module.

Table 14-64. JCM_EN

Value	Description
0x0	Digital clock jitter module is disabled.
0x1	Digital clock jitter module is enabled.

1. If the Antenna Diversity algorithm is enabled (ANT_DIV_EN = 1), the value shall be set to PDT_THRES = 3, otherwise it shall be set back to the reset value. This is not automatically done by the hardware.

14.8.11 SFD_VALUE

Name: SFD_VALUE
Offset: 0x0B
Reset: 0xA7
Property: -

The SFD_VALUE register contains the one octet start-of-frame delimiter (SFD).

Bit	7	6	5	4	3	2	1	0
	SFD_VALUE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	0	1	1	1

Bits 7:0 – SFD_VALUE[7:0] SFD_VALUE
The register bits SFD_VALUE are required for transmit and receive operation.

Table 14-65. SFD_VALUE

Value	Description
0xA7	For transmission this value is copied into start-of-frame delimiter (SFD) field of frame header. For reception this value is checked for incoming frames. The default value is according to IEEE 802.15.4 specification.

For IEEE 802.15.4 compliant networks, set SFD_VALUE = 0xA7 as specified in [2]. This is the default value of the register.
To establish non IEEE 802.15.4 compliant networks, the SFD value can be changed to any other value. If enabled, IRQ_2 (RX_START) is issued only if the received SFD matches SFD_VALUE and a valid PHR is received.

14.8.12 TRX_CTRL_2

Name: TRX_CTRL_2
Offset: 0x0C
Reset: 0x14
Property: -

The TRX_CTRL_2 register is a multi-purpose control register to control various settings of the radio transceiver.

Bit	7	6	5	4	3	2	1	0
	RX_SAFE_MODE	TRX_OFF_AVDD_EN	OQPSK_SCRAM_EN	ALT_SPECTRUM	BPSK_OQPSK	SUB_MODE	OQPSK_DATA_RATE[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	0	0

Bit 7 – RX_SAFE_MODE RX_SAFE_MODE
Protect Frame Buffer after frame reception with valid FCF check.

Table 14-66. RX_SAFE_MODE

Value	Description
0x0	Disable Dynamic Frame Buffer protection
0x1(1)	Enable Dynamic Frame Buffer protection

1. Dynamic Frame Buffer Protection is released on the rising edge of pin 23 (/SEL) during a Frame Buffer read access, or on the radio transceiver's state change from RX_ON or RX_AACK_ON to another state.

This operation mode is independent of the setting of the RX_PDT_LEVEL bits in the RX_SYN register.

Bit 6 – TRX_OFF_AVDD_EN TRX_OFF_AVDD_EN
The register bit TRX_OFF_AVDD_EN enables analog voltage regulator in TRX_OFF state.

Table 14-67. TRX_OFF_AVDD_EN

Value	Description
0x0	During TRX_OFF state analog voltage regulator is disabled.
0x1	During TRX_OFF state analog voltage regulator is enabled.

If this register bit is set, the analog voltage regulator remains enabled in TRX_OFF state. This provides for a faster RX or TX turn-on time. It is especially useful when a short stopover is made in TRX_OFF state. The recharge time for capacitances is avoided in this case.
The current consumption increases by 100µA (typical).

Bit 5 – OQPSK_SCRAM_EN OQPSK_SCRAM_EN
If register bit OQPSK_SCRAM_EN is enabled, an additional chip scrambling for O-QPSK is applied for data rate 400kb/s and 1000kb/s.

Table 14-68. OQPSK_SCRAM_EN

Value	Description
0x0	Scrambler is disabled
0x1	Scrambler is enabled

Bit 4 – ALT_SPECTRUM ALT_SPECTRUM
The register bit ALT_SPECTRUM controls an alternative spectrum for different modes.

Table 14-69. ALT_SPECTRUM

Value	Description
0x0	The alternative spectrum mode is disabled
0x1	The alternative spectrum mode is enabled

BPSK with 40kb/s: If set to zero, a chip sequence according to IEEE 802.15.4 is used. If set to one, a modified chip sequence interoperable with IEEE 802.15.4 is used for TX and RX showing different spectrum properties (to ensure FCC 600kHz bandwidth requirement). This might be beneficial when using an external power amplifier and targeting high output power according to FCC 15.247 [5].
O-QPSK with 400kchip/s: If set to zero, pulse shaping is a combination of half-sine shaping and RC-0.2 shaping according to IEEE 802.15.4. If set to one, pulse shaping is RC-0.2 shaping. This avoids inter-chip interference which results in a significantly lower EVM, refer to Section 12.6. The peak to average ratio increases by about 1dB.
O-QPSK with 1000kchip/s: If set to zero, pulse shaping is half-sine shaping. If set to one, pulse shaping is RC-0.8 shaping. Compared with half-sine shaping, side-lobes are reduced at the expense of an increased peak to average ratio (~1dB); refer toFigure 9-11 and Figure 9-12, respectively. This mode is particularly suitable for the Chinese 780MHz band, refer to IEEE 802.15.4c-2009 [3] or IEEE 802.15.4-2011 [4].
Notes:

1. The modulation BPSK-40 and modulation BPSK-40-ALT are interoperable together, with some performance degenerations.
2. During reception, this bit is not evaluated within the AT86RF212B, so it is not explicitly required to align different transceivers with ALT_SPECTRUM in order to assure interoperability. It is very likely that this also holds for any IEEE 802.15.4-2006 compliant O-QPSK transceiver in the 915MHz band, since the IEEE 802.15.4-2006 requirements are fulfilled for both types of shaping.

Bit 3 – BPSK_OQPSK BPSK_OQPSK
The register bit BPSK_OQPSK controls the modulation scheme.

Table 14-70. BPSK_OQPSK

Value	Description
0x0	BPSK modulation is active
0x1	O-QPSK modulation is active

Bit 2 – SUB_MODE SUB_MODE
Mode selection for European/North American/(Chinese) band.

Table 14-71. SUB_MODE

Value	Description
0x0	BPSK-20, OQPSK-{100,200,400}
0x1	BPSK-40, OQPSK-{250,500,1000}

If set to one (reset value), the chip rate is 1000kchip/s for BPSK_OQPSK = 1 and 600kchip/s for BPSK_OQPSK = 0. It permits data rates out of {250, 500, 1000}kb/s or 40kb/s, respectively. This mode is particularly suitable for the 915MHz band. For O-QPSK transmission, pulse shaping is either half-sine shaping or RC-0.8 shaping, depending on ALT_SPECTRUM.
If set to zero, the chip rate is 400kchip/s for BPSK_OQPSK = 1 and 300kchip/s for BPSK_OQPSK = 0. It permits data rates out of {100, 200, 400}kb/s or 20kb/s, respectively. This mode is particularly suitable for the 868.3MHz band. For O-QPSK transmission, pulse shaping is always the combination of half-sine shaping and RC-0.2 shaping.

Bits 1:0 – OQPSK_DATA_RATE[1:0] OQPSK_DATA_RATE
A write access to these register bits set the O-QPSK PSDU data rate used by the radio transceiver. The reset value OQPSK_DATA_RATE = 0 is the PSDU data rate according to IEEE 802.15.4.

Table 14-72. OQPSK_DATA_RATE

Value	Description
0x0	SUB_MODE = 0: 100kb/s or SUB_MODE = 1: 250kb/s
0x1	SUB_MODE = 0: 200kb/s or SUB_MODE = 1: 500kb/s
0x2	SUB_MODE = 0: 400kb/s or SUB_MODE = 1: 1000kb/s

.....continued	
Value	**Description**
0x3	SUB_MODE = 0: Reserved or SUB_MODE = 1: 500kb/s

The AT86RF212B supports two different modes with an PSDU data rate of 500kb/s. Using OQPSK_DATA_RATE = 3 might be beneficial when using an external power amplifier and targeting high output power according to FCC 15.247 [5].
In the table below all PHY modes supported by the AT86RF212B are summarized with the relevant setting for each bit of register TRX_CTRL_2. The “-“ (minus) character means that the bit entry is not relevant for the particular PHY mode.

	TRX_CTRL_2 Register Bits								
PHY Mode	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Compliance**
BPSK-20	-	-	-	0	0	0	0	0	IEEE 802.15.4-2003/2006/2011: channel page 0, channel 0
BPSK-40	-	-	-	0	0	1	0	0	IEEE 802.15.4-2003/2006/2011: channel page 0, channel 1 to 10
BPSK-40-ALT	-	-	-	1	0	1	0	0	Proprietary, alternative spreading code
OQPSK-SIN-RC-100	-	-	-	0	1	0	0	0	IEEE 802.15.4-2006/2011: channel page 2, channel 0
OQPSK-SIN-RC-200	-	-	-	0	1	0	0	1	Proprietary
OQPSK-SIN-RC-400-SCR-ON	-	-	1	0	1	0	1	0	Proprietary, scrambler on
OQPSK-SIN-RC-400-SCR-OFF	-	-	0	0	1	0	1	0	Proprietary, scrambler off
OQPSK-RC-100	-	-	-	1	1	0	0	0	Proprietary
OQPSK-RC-200	-	-	-	1	1	0	0	1	Proprietary
OQPSK-RC-400-SCR-ON	-	-	1	1	1	0	1	0	Proprietary, scrambler on
OQPSK-RC-400-SCR-OFF	-	-	0	1	1	0	1	0	Proprietary, scrambler off
OQPSK-SIN-250	-	-	-	0	1	1	0	0	IEEE 802.15.4-2006/2011: channel page 2, channel 1 to 10
OQPSK-SIN-500	-	-	-	0	1	1	0	1	Proprietary
OQPSK-SIN-500-ALT	-	-	-	0	1	1	1	1	Proprietary, alternative spreading code
OQPSK-SIN-1000-SCR-ON	-	-	1	0	1	1	1	0	Proprietary, scrambler on
OQPSK-SIN-1000-SCR-OFF	-	-	0	0	1	1	1	0	Proprietary, scrambler off
OQPSK-RC-250	-	-	-	1	1	1	0	0	IEEE 802.15.4-2011: channel page 5, channel 0 to 3
OQPSK-RC-500	-	-	-	1	1	1	0	1	Proprietary
OQPSK-RC-500-ALT	-	-	-	1	1	1	1	1	Proprietary, alternative spreading code
OQPSK-RC-1000-SCR-ON	-	-	1	1	1	1	1	0	Proprietary, scrambler on
OQPSK-RC-1000-SCR-OFF	-	-	0	1	1	1	1	0	Proprietary, scrambler off

14.8.13 ANT_DIV

Name: ANT_DIV
Offset: 0x0D
Reset: 0x01
Property: -

The ANT_DIV register controls Antenna Diversity.

Bit	7	6	5	4	3	2	1	0
	ANT_SEL				ANT_DIV_EN	ANT_EXT_SW_EN	ANT_CTRL[1:0]	
Access	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit 7 – ANT_SEL ANT_SEL
Signals status of antenna at the time of the last IRQ_2 (RX_START) interrupt, IRQ_3 (TRX_END) interrupt, or TX_START event.

Table 14-73. ANT_SEL

Value	Description
0x0	Antenna 0
0x1	Antenna 1

The register bit signals the status of the selected antenna at the time of the last IRQ_2 (RX_START) interrupt, IRQ_3 (TRX_END) interrupt, or TX_START event. This information can be used to build up a history of the antenna used for successful transmission (indicated by an acknowledgement frame) in TX_ARET mode.

Bit 3 – ANT_DIV_EN ANT_DIV_EN
The register bit ANT_DIV_EN controls TX antenna diversity.

Table 14-74. ANT_DIV_EN

Value	Description
0x0	TX antenna diversity is disabled.
0x1	TX antenna diversity is enabled.

If set to one, antenna diversity is enabled in TX_ARET mode with expected ACK reply. The transceiver automatically selects the antenna with the aim to minimize the number of retransmissions.

Bit 2 – ANT_EXT_SW_EN ANT_EXT_SW_EN
The register bit ANT_EXT_SW_EN controls the external antenna switch.

Table 14-75. ANT_EXT_SW_EN

Value	Description
0x0	Antenna Diversity RF switch control is disabled
0x1	Antenna Diversity RF switch control is enabled

If enabled, pin 9 (DIG1) and pin 10 (DIG2) become output pins and provide a differential control signal for an antenna diversity switch. The selection of an antenna within TX_ARET mode is done automatically if ANT_DIV_EN = 1, or, if ANT_DIV_EN = 0, according to register bits ANT_CTRL.
If RX Frame Time Stamping is used in combination with Antenna Diversity, pin 9 (DIG1) is used for Antenna Diversity and pin 10 (DIG2) is used for RX Frame Time Stamping. AT86RF212B does not provide a differential control signal in this case.
If the register bit is set, the control pins DIG1/DIG2 are activated in all radio transceiver states as long as register bit ANT_EXT_SW_EN is set. If the AT86RF212B is not in a receive or transmit state, it is recommended to disable register bit ANT_EXT_SW_EN to reduce the power consumption or avoid leakage current of an external RF switch,

especially during SLEEP state. If register bit ANT_EXT_SW_EN = 0, output pins DIG1 and DIG2 are internally connected to digital ground.

Bits 1:0 – ANT_CTRL[1:0] ANT_CTRL
These register bits provide a static control of an Antenna Diversity switch.

Table 14-76. ANT_CTRL

Value	Description
Reserved	Reserved
0x1	Antenna 0 DIG1 = L DIG2 = H
0x2	Antenna 1 DIG1 = H DIG2 = L
Reserved	Reserved

These register bits provide a static control of an Antenna Diversity switch if ANT_DIV_EN = 0 and ANT_EXT_SW_EN = 1. Although it is possible to change register bits ANT_CTRL in state TRX_OFF, this change will be effective at DIG1 and DIG2 in states PLL_ON and RX_ON.

14.8.14 IRQ_MASK

Name: IRQ_MASK
Offset: 0x0E
Reset: 0x00
Property: -

The IRQ_MASK register controls the interrupt signaling via the IRQ.

Bit	7	6	5	4	3	2	1	0
	IRQ_MASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IRQ_MASK[7:0] IRQ_MASK
Mask register for interrupts. IRQ_MASK[7] correspondents to IRQ_7 (BAT_LOW). IRQ_MASK[0] correspondents to IRQ_0 (PLL_LOCK).

Table 14-77. IRQ_MASK

Value	Description
0x00	The IRQ_MASK register is used to enable or disable individual interrupts. An interrupt is enabled if the corresponding bit is set to one. All interrupts are disabled after power-on sequence (P_ON state) or reset (RESET state). Valid values are [0xFF, 0xFE, …, 0x00].

1. If an interrupt is enabled it is recommended to read the interrupt status register 0x0F (IRQ_STATUS) first to clear the history.

14.8.15 IRQ_STATUS

Name: IRQ_STATUS
Offset: 0x0F
Reset: 0x00
Property: -

The IRQ_STATUS register contains the status of the pending interrupt requests.

By reading the register after an interrupt is signaled by the IRQ signal to the microcontroller the source of the issued interrupt can be identified. A read access to this register resets all interrupt bits, and so clears the IRQ_STATUS register.

Notes:

1. If the IRQ_MASK_MODE bit in the TRX_CTRL_1 register is set, an interrupt event can be read from IRQ_STATUS register even if the interrupt itself is masked; refer to Figure 14-18. However in that case no timing information for this interrupt is provided.
2. If the IRQ_MASK_MODE bit in the TRX_CTRL_1 is set, it is recommended to read the interrupt status register (IRQ_STATUS) first to clear the history.

Bit	7	6	5	4	3	2	1	0
	IRQ_7_BAT_LOW	IRQ_6_TRX_UR	IRQ_5_AMI	IRQ_4_CCA_ED_DONE	IRQ_3_TRX_END	IRQ_2_RX_START	IRQ_1_PLL_UNLOCK	IRQ_0_PLL_LOCK
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – IRQ_7_BAT_ LOW IRQ_7_BAT_ LOW
Indicates a supply voltage below the programmed threshold.

Bit 6 – IRQ_6_TRX_ UR IRQ_6_TRX_ UR
Indicates a Frame Buffer access violation.

Bit 5 – IRQ_5_AMI IRQ_5_AMI
Indicates address matching.

Bit 4 – IRQ_4_ CCA_ED_ DONE IRQ_4_ CCA_ED_ DONE
Multi-functional interrupt:

1. AWAKE_END: Indicates finished transition to TRX_OFF state from P_ON, SLEEP, DEEP_SLEEP, or RESET state.
2. CCA_ED_DONE: Indicates the end of a CCA or ED measurement.

Bit 3 – IRQ_3_TRX_ END IRQ_3_TRX_ END
RX: Indicates the completion of a frame reception.
TX: Indicates the completion of a frame transmission.

Bit 2 – IRQ_2_RX_ START IRQ_2_RX_ START
Indicates the start of a PSDU reception; the AT86RF233 state changed to BUSY_RX; the PHR can be read from Frame Buffer.

Bit 1 – IRQ_1_PLL_ UNLOCK IRQ_1_PLL_ UNLOCK
Indicates PLL unlock. If the radio transceiver is in BUSY_TX / BUSY_TX_ARET state, the PA is turned off immediately.

Bit 0 – IRQ_0_PLL_ LOCK IRQ_0_PLL_ LOCK
Indicates PLL lock.

14.8.16 VREG_CTRL

Name: VREG_CTRL
Offset: 0x10
Reset: 0x00
Property: -

The VREG_CTRL register controls the use of the voltage regulators and indicates the status of these.

Bit	7	6	5	4	3	2	1	0
	AVREG_EXT	AVDD_OK			DVREG_EXT	DVDD_OK		
Access	R/W	R	R	R	R/W	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – AVREG_EXT AVREG_EXT
If set this register bit disables the internal analog voltage regulator to apply an external regulated 1.8V supply for the analog building blocks.

Table 14-78. AVREG_EXT

Value	Description
0x0	Internal voltage regulator enabled, analog section
0x1	Internal voltage regulator disabled, use external regulated 1.8V supply voltage for the analog section

Bit 6 – AVDD_OK AVDD_OK
This register bit indicates if the internal 1.8V regulated voltage supply AVDD has settled. The bit is set to logic high, if AVREG_EXT = 1.

Table 14-79. AVDD_OK

Value	Description
0x0	Analog voltage regulator is disabled or supply voltage not stable
0x1	Analog supply voltage is stable

Bit 3 – DVREG_EXT DVREG_EXT
If set this register bit disables the internal digital voltage regulator to apply an external regulated 1.8V supply for the digital building blocks.

Table 14-80. DVREG_EXT

Value	Description
0x0	Internal voltage regulator enabled, digital section
0x1	Internal voltage regulator disabled, use external regulated 1.8V supply voltage for the digital section

Bit 2 – DVDD_OK DVDD_OK
This register bit indicates if the internal 1.8V regulated voltage supply DVDD has settled. The bit is set to logic high, if DVREG_EXT = 1.

Table 14-81. DVDD_OK

Value	Description
0x0	Digital voltage regulator is disabled or supply voltage not stable
0x1	Digital supply voltage is stable

1. While the reset value of this bit is zero, any practical access to the register is only possible when DVREG is active. So this bit is normally always read out as one.

14.8.17 BATMON

Name: BATMON
Offset: 0x11
Reset: 0x02
Property: -

The BATMON register configures the battery monitor to compare the supply voltage at EVDD to the threshold. Additionally, the supply voltage status at EVDD can be read from register bit BATMON_OK according to the actual BATMON settings.

Bit	7	6	5	4	3	2	1	0
	PLL_LOCK_CP		BATMON_OK	BATMON_HR	BATMON_VTH[3:0]			
Access	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit 7 – PLL_LOCK_CP PLL_LOCK_CP
The register bit PLL_LOCK_CP signals the current status of the PLL lock comparator output.

Table 14-82. PLL_LOCK_CP

Value	Description
0x0	PLL is currently unlocked
0x1	PLL is currently locked

Bit 5 – BATMON_OK BATMON_OK
The register bit BATMON_OK indicates the level of the external supply voltage with respect to the programmed threshold BATMON_VTH.

Table 14-83. BATMON_OK

Value	Description
0x0	The battery voltage is below the threshold
0x1	The battery voltage is above the threshold

Bit 4 – BATMON_HR BATMON_HR
The register bit BATMON_HR sets the range and resolution of the battery monitor.

Table 14-84. BATMON_HR

Value	Description
0x0	Enables the low range, see BATMON_VTH
0x1	Enables the high range, see BATMON_VTH

Bits 3:0 – BATMON_VTH[3:0] BATMON_VTH
The voltage threshold values for the battery monitor are set by register bits BATMON_VTH.

Table 14-85. BATMON_VTH

Value BATMON_VTH	Voltage [V] BATMON_HR = 1	Voltage [V] BATMON_HR = 0
0x0	2.550	1.70
0x1	2.625	1.75
0x2	2.700	1.80
0x3	2.775	1.85
0x4	2.850	1.90
0x5	2.925	1.95
0x6	3.000	2.00
0x7	3.075	2.05

.....continued		
Value BATMON_VTH	Voltage [V] BATMON_HR = 1	Voltage [V] BATMON_HR = 0
0x8	3.150	2.10
0x9	3.225	2.15
0xA	3.300	2.20
0xB	3.375	2.25
0xC	3.450	2.30
0xD	3.525	2.35
0xE	3.600	2.40
0xF	3.675	2.45

14.8.18 XOSC_CTRL

Name: XOSC_CTRL
Offset: 0x12
Reset: 0xF0
Property: -

The XOSC_CTRL register controls the operation of the crystal oscillator.

Bit	7	6	5	4	3	2	1	0
	XTAL_MODE[3:0]				XTAL_TRIM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	0	0	0	0

Bits 7:4 – XTAL_MODE[3:0] XTAL_MODE
The register bits XTAL_MODE sets the operating mode of the crystal oscillator.

Table 14-86. XTAL_MODE

Value	Description
0x4	Internal crystal oscillator disabled, use external reference frequency
0xF	Internal crystal oscillator enabled and XOSC voltage regulator enabled
	All other values are reserved

For normal operation the default value is set to XTAL_MODE = 0xF after reset. Using an external clock source it is recommended to set XTAL_MODE = 0x4.

Bits 3:0 – XTAL_TRIM[3:0] XTAL_TRIM
The register bits XTAL_TRIM control internal capacitance arrays connected to the XTAL1 and XTAL2 pins.

Table 14-87. XTAL_TRIM

Value	Description
0x0	A capacitance value in the range from 0pF to 4.5pF is selectable with a resolution of 0.3pF. Valid values are [0xF, 0xE, …, 0x0].

14.8.19 CC_CTRL_0

Name: CC_CTRL_0
Offset: 0x13
Reset: 0x00
Property: -

The CC_CTRL_0 register controls the frequency selection, if the selection by CHANNEL (register 0x08, PHY_CC_CCA) is not used.

Bit	7	6	5	4	3	2	1	0
	CC_NUMBER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CC_NUMBER[7:0] CC_NUMBER
These bitscontrols the center frequency if the selection by channel number according to IEEE 802.15.4 is not used.

Table 14-88. CC_NUMBER

Value	Description
0x00	Alternative frequency selection with 100kHz or 1MHz frequency spacing CC_BAND = 0x0: Not used CC_BAND = 0x1: Valid values are [0xFF, 0xFE, …,0x00] CC_BAND = 0x2: Valid values are [0xFF, 0xFE, …,0x00] CC_BAND = 0x3: Valid values are [0xFF, 0xFE, …,0x00] CC_BAND = 0x4: Valid values are [0x5E, 0x5D, …,0x00] CC_BAND = 0x5: Valid values are [0x66, 0x65, …,0x00] CC_BAND = 0x6: Valid values are [0xFF, 0xFE, …,0x00] All other values are reserved

14.8.20 CC_CTRL_1

Name: CC_CTRL_1
Offset: 0x14
Reset: 0x00
Property: -

The CC_CTRL_1 register controls the selection of the frequency bands.

Bit	7	6	5	4	3	2	1	0
						CC_BAND[2:0]		
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 2:0 – CC_BAND[2:0] CC_BAND
The register bits CC_BAND control the selection for IEEE 802.15.4 channel band and additional frequencies bands.

Table 14-89. CC_BAND

Value	Description
0x0	The IEEE 802.15.4 channel within register bits CHANNEL is selected
0x1	The frequency band one is selected
0x2	The frequency band two is selected
0x3	The frequency band three is selected
0x4	The frequency band four is selected
0x5	The frequency band five is selected
0x6	The frequency band six is selected
Reserved	All other values are reserved

If the register bits CC_BAND and CC_NUMBER are used, the frequency mapping is described in the table below.

Table 14-90. Frequency Bands and Numbers

CC_BAND	CC_NUMBER	Description
0x0	Not used	European and North American channels according to IEEE 802.15.4; Frequency selected by register bits CHANNEL (register 0x08, PHY_CC_CCA), refer to Section 9.8.6
0x1	0x00 – 0xFF	769.0MHz – 794.5MHz F_C [MHz] = 769.0[MHz] + 0.1[MHz] x CC_NUMBER
0x2	0x00 – 0xFF	857.0MHz – 882.5MHz F_C [MHz] = 857.0[MHz] + 0.1[MHz] x CC_NUMBER
0x3	0x00 – 0xFF	903.0MHz – 928.5MHz F_C[MHz] = 903.0[MHz] + 0.1[MHz] x CC_NUMBER
0x4	0x00 – 0x5F	769MHz – 863MHz F_C [MHz] = 769[MHz] + 1[MHz] x CC_NUMBER
0x5	0x00 – 0x66	833MHz – 935MHz F_C [MHz] = 833[MHz] + 1[MHz] x CC_NUMBER
0x6	0x00 – 0xFF	902.0MHz – 927.5MHz F_C [MHz] = 902.0[MHz] + 0.1[MHz] x CC_NUMBER
0x7	0x00 – 0xFF	Reserved

14.8.21 RX_SYN

Name: RX_SYN
Offset: 0x15
Reset: 0x00
Property: -

The register RX_SYN controls the blocking of receiver path and the sensitivity threshold of the receiver.

Bit	7	6	5	4	3	2	1	0
	RX_PDT_DIS	RX_OVERRIDE[2:0]			RX_PDT_LEVEL[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – RX_PDT_DIS RX_PDT_DIS
The register bit RX_PDT_DIS prevents the reception of a frame during RX phase.

Table 14-91. RX_PDT_DIS

Value	Description
0x0	RX path is enabled
0x1	RX path is disabled

RX_PDT_DIS = 1 prevents the reception of a frame even if the radio transceiver is in receive modes. An ongoing frame reception is not affected. This operation mode is independent of the setting of register bits RX_PDT_LEVEL.

Bits 6:4 – RX_OVERRIDE[2:0] RX_OVERRIDE
The register bits RX_OVERRIDE control the RXO functions during RX phase. During the receive process the validity of the current frame and the occurrence of a strong interferer is checked continuously. In either of those cases the reception is automatically restarted to increase the overall system availability and throughput with respect to correct received packets.

Table 14-92. RX_OVERRIDE

Value	Description
0x0(1)	All RX override functions are disabled (default)
0x6(2)	IPAN scanning is enabled, 9dB Energy Detection (ED) check is enabled, Link Quality (LQ) check is enabled
Reserved	All other values are reserved

1. Frames are decoded up to the length specified in the PHR field, independent of any interference while receiving this frame.
2. Detection of strong interference while receiving a frame (where the frame is destroyed anyway) and fast re-synchronization to a potential new frame.
3. There is no TRX_END interrupt for an ongoing frame reception when re-synchronization is forced by strong interference.

The Receiver Override can be used without performance degradation in combination with any modulation scheme and data rate.

Bits 3:0 – RX_PDT_LEVEL[3:0] RX_PDT_LEVEL
The register bits RX_PDT_LEVEL desensitize the receiver in steps of 3dB.

Table 14-93. RX_PDT_LEVEL

Value	Description
0x00	Maximum RX sensitivity
0x0F	RX input level[dBm] > RSSI_BASE_VAL[dBm] + 3[dB] x 14

These register bits desensitize the receiver such that frames with an RSSI level below the RX_PDT_LEVEL threshold level (if RX_PDT_LEVEL > 0) are not received. For a RX_PDT_LEVEL > 0 value the threshold level can be calculated according to the following formula:
P_{RF}[dBm] > $RSSI_{BASE\_VAL}$[dBm] + 3[dB] x (RX_PDT_LEVEL - 1)
If register bits RX_PDT_LEVEL = 0 (reset value) all frames with a valid SHR and PHR are received, independently of their signal strength.
If register bits RX_PDT_LEVEL > 0, the current consumption of the receiver in all RX listening states is reduced by 500µA.

14.8.22 RF_CTRL_0

Name: RF_CTRL_0
Offset: 0x16
Reset: 0x31
Property: -

The register RF_CTRL_0 contains control settings to configure the transmit path.

Bit	7	6	5	4	3	2	1	0
	PA_LT[1:0]						GC_TX_OFFS[1:0]	
Access	R/W	R/W					R/W	R/W
Reset	0	0	1	1	0	0	0	1

Bits 7:6 – PA_LT[1:0] PA_LT
This bit control lead time of the PA (relative to first chip of TX data).

Table 14-94. PA_LT

Value	Description
0x0	2µs
0x1	4µs
0x2	6µs
0x3	8µs

These register bits control the lead time of the PA enable signal relative to the TX data start. This allows to enable the PA 2, 4, 6, or 8µs before the transmit signal starts. The PA enable signal can also be output at differential pin pair DIG3/DIG4 to provide a control signal for an external RF front-end.

Bits 1:0 – GC_TX_OFFS[1:0] GC_TX_OFFS
The register bits GC_TX_OFFS control the TX power offset.

Table 14-95. GC_TX_OFFS

Value	Description
0x0	Reserved
0x1	0dB
0x2	+1dB
0x3	+2dB

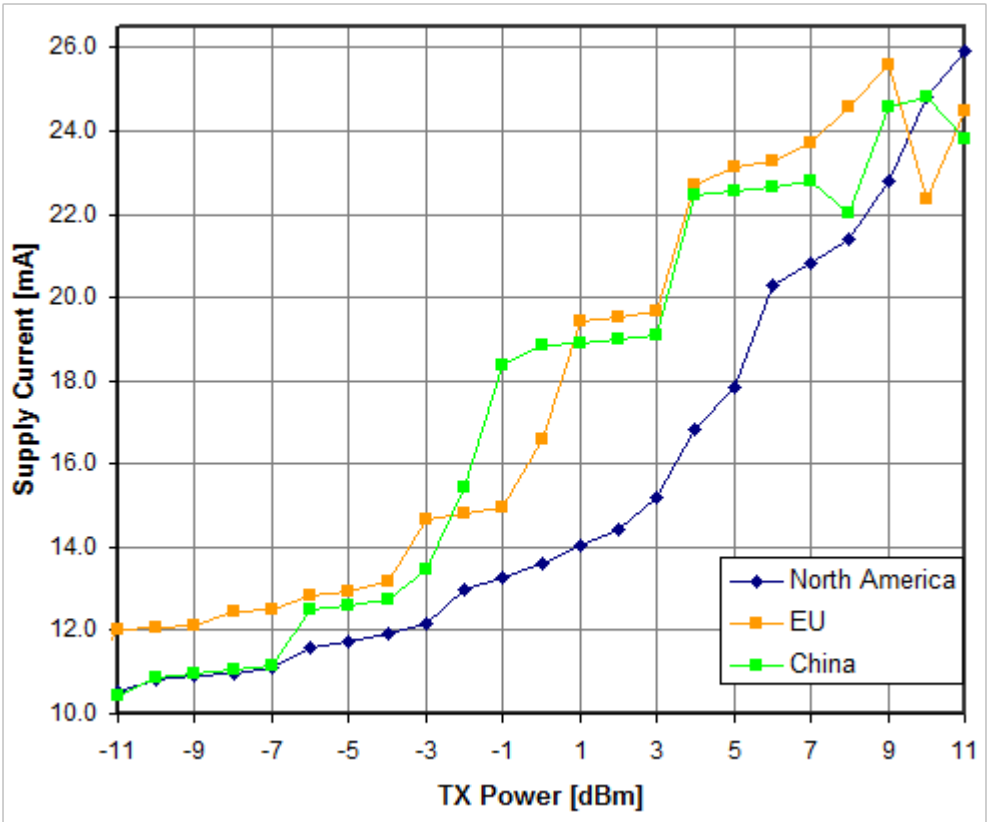
These register bits provide an offset between the TX power control word TX_PWR (register 0x05, PHY_TX_PWR) and the actual TX power. This 2-bit word is added to the TX power control word before it is applied to the circuit block which adjusts the TX power. It can be used to compensate differences of the average TX power depending of the modulation format.

Table 14-96. Mode-Dependent Setting of GC_TX_OFFS

Mode	BPSK	O-QPSK
GC_TX_OFFS	0x3	0x2

Exception for OQPSK-RC-{100,200,400}, see Table 14-54

Figure 14-66. Supply Currents for O-QPSK Modulation depending on TX Power Setting



The North American mapping table is optimized for lowest supply current. The more relaxed spectral side lobe requirements of the IEEE 802.15.4 standard are fulfilled.
The European mapping tables take into account that linearity is needed to keep the out-of-band spurious emissions below the ETSI requirements, refer to [6]. Regulatory requirements with respect to power density (depending on the frequency band used) are not considered, refer to [7].
The European mapping takes more supply current than the North American map and uses the normal (linearized) PA mode to provide medium output power up to 3dBm for OQPSK-SIN-RC-{100/200/400} modes and 6dBm for BPSK-20 mode.
The Chinese mapping uses the boost mode to provide higher TX power levels at the expense of higher supply current. As a result, the maximum TX power is 11dBm for OQPSK-RC-{250/500) modes.
Due to great regional distinctions of regulatory requirements, it is not possible to cover all restrictions in this datasheet. Manufactures must take the responsibility to check measurement results against the latest regulations of nations into which they market.

14.8.23 XAH_CTRL_1

Name: XAH_CTRL_1
Offset: 0x17
Reset: 0x00
Property: -

The XAH_CTRL_1 register is a multi-purpose controls register for Extended Operating Mode.

Bit	7	6	5	4	3	2	1	0
		CSMA_LBT_MODE	AACK_FLTR_RES_FT	AACK_UPLD_RES_FT		AACK_ACK_TIME	AACK_PROM_MODE	
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit 6 – CSMA_LBT_MODE CSMA_LBT_MODE
The register bit CSMA_LBT_MODE switched between CSMA-CA or Listen Before Talk (LBT) algorithm within TX_ARET mode.

Table 14-97. CSMA_LBT_MODE

Value	Description
0x0	CSMA-CA alogrithm is used.
0x01	LBT algorithm is used

If set to zero (default), CSMA-CA algorithm is used during TX_ARET for clear channel assessment. Otherwise, the LBT specific listening mode is applied.

Bit 5 – AACK_FLTR_RES_FT AACK_FLTR_RES_FT
Filter reserved frame types like data frame type. The register bit AACK_FLTR_RES_FT shall only be set if register bit AACK_UPLD_RES_FT = 1.

Table 14-98. AACK_FLTR_RES_FT

Value	Description
0x0(1)	Filtering reserved frame types is disabled
0x1(2)	Filtering reserved frame types is enabled

1. If AACK_FLTR_RES_FT = 0 the received reserved frame is only checked for a valid FCS.
2. If AACK_FLTR_RES_FT = 1 reserved frame types are filtered similar to data frames as specified in IEEE 802.15.4–2006.

Reserved frame types are explained in IEEE 802.15.4 Section 7.2.1.1.1.

Bit 4 – AACK_UPLD_RES_FT AACK_UPLD_RES_FT
Upload reserved frame types within RX_AACK mode.

Table 14-99. AACK_UPLD_RES_FT

Value	Description
0x0	Upload of reserved frame types is disabled
0x1(1)	Upload of reserved frame types is enabled

1. If AACK_UPLD_RES_FT = 1 received frames indicated as a reserved frame are further processed. For those frames, an IRQ_3 (TRX_END) interrupt is generated if the FCS is valid.

In conjunction with the configuration bit AACK_FLTR_RES_FT, these frames are handled like IEEE 802.15.4 compliant data frames during RX_AACK transaction. An IRQ_5 (AMI) interrupt is issued, if the addresses in the received frame match the node's addresses.

That means, if a reserved frame passes the third level filter rules, an acknowledgement frame is generated and transmitted if it was requested by the received frame. If this is not wanted register bit AACK_DIS_ACK (register 0x2E, CSMA_SEED_1) has to be set.

Bit 2 – AACK_ACK_TIME AACK_ACK_TIME
The register bit AACK_ACK_TIME controls the acknowledgment frame response time within RX_AACK mode.

Table 14-100. AACK_ACK_TIME

Value	Description
0x0	Acknowledgment time is 12 symbol periods (a Turnaround Time)
0x1	Two symbol periods: BPSK-20, OQPSK-{100,200,400}; Three symbol periods: BPSK-40, OQPSK-{250,500,1000}

According to IEEE 802.15.4-2006, Section 7.5.6.4.2 the transmission of an acknowledgment frame shall commence 12 symbol periods (aTurnaroundTime) after the reception of the last symbol of a data or MAC command frame. This is achieved with the reset value of the register bit AACK_ACK_TIME.
Alternatively, if AACK_ACK_TIME = 1, the acknowledgment response time is reduced according to the table below.

Table 14-101. Short ACK Response Time (AACK_ACK_TIME=1)

PHY Mode	ACK Response Time [symbol periods]
BPSK-20, OQPSK-{100,200,400}	2
BPSK-40, OQPSK-{250,500,1000}	3

The reduced ACK response time is particularly useful for the High Data Rate Modes.

Bit 1 – AACK_PROM_MODE AACK_PROM_MODE
The register bit AACK_PROM_MODE enables the promiscuous mode, within the RX_AACK mode.

Table 14-102. AACK_PROM_MODE

Value	Description
0x0	Promiscuous mode is disabled
0x1	Promiscuous mode is enabled

Refer to [2] IEEE 802.15.4-2006 Section 7.5.6.5.
If this register bit is set, every incoming frame with a valid PHR finishes with IRQ_3 (TRX_END) interrupt even if the third level filter rules do not match or the FCS is not valid. However, register bit RX_CRC_VALID (register 0x06, PHY_RSSI) is set accordingly.
In contrast to [2] IEEE 802.15.4-2006, if a frame passes the third level filter rules, an acknowledgement frame is generated and transmitted unless disabled by register bit AACK_DIS_ACK (register 0x2E, CSMA_SEED_1), or use Basic Operating Mode instead.

14.8.24 FTN_CTRL

Name: FTN_CTRL
Offset: 0x18
Reset: 0x58
Property: -

The FTN_CTRL register controls the operation of the filter tuning network calibration loop.

Bit	7	6	5	4	3	2	1	0
	FTN_START							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	1	0	0	0

Bit 7 – FTN_START FTN_START
Manual start of a filter calibration cycle.

Table 14-103. FTN_START

Value	Description
0x0	Filter calibration is finished
0x1	Initiates filter calibration cycle

FTN_START = 1 initiates the filter tuning network calibration. When the calibration cycle has finished after t_{FTN} = 25µs (typ.). The register bit is cleared immediately after finishing the calibration.

14.8.25 PLL_CF

Name: PLL_CF
Offset: 0x1A
Reset: 0x48
Property: -

The PLL_CF register controls the operation of the center frequency calibration loop.

Bit	7	6	5	4	3	2	1	0
	PLL_CF_START							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	0	0	0

Bit 7 – PLL_CF_START PLL_CF_START
Manual start of center frequency calibration cycle.

Table 14-104. PLL_CF_START

Value	Description
0x0	Center frequency calibration cycle is finished
0xF	Initiates center frequency calibration cycle

PLL_CF_START = 1 initiates the center frequency calibration. The calibration cycle has finished after t_{PLL_CF} = 8µs (typ.). The register bit is cleared immediately after finishing the calibration.

14.8.26 PLL_DCU

Name: PLL_DCU
Offset: 0x1B
Reset: 0x40
Property: -

The PLL_DCU register controls the operation of the delay cell calibration loop.

Bit	7	6	5	4	3	2	1	0
	PLL_DCU_START							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit 7 – PLL_DCU_START PLL_DCU_START
Manual start of delay cell calibration cycle.

Table 14-105. PLL_DCU_START

Value	Description
0x0	Delay cell calibration cycle is finished
0x1	Initiates delay cell calibration cycle

PLL_DCU_START = 1 initiates the delay cell calibration. The calibration cycle has finished after t_{PLL_DCU} = 10µs. The register bit is cleared immediately after finishing the calibration.

14.8.27 PART_NUM

Name: PART_NUM
Offset: 0x1C
Reset: 0x07
Property: -

The register PART_NUM can be used for the radio transceiver identification and includes the part number of the device.

Bit	7	6	5	4	3	2	1	0
	PART_NUM[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bits 7:0 – PART_NUM[7:0] PART_NUM

Table 14-106. PART_NUM

Value	Description
0x07	AT86RF212B part number

14.8.28 VERSION_NUM

Name: VERSION_NUM
Offset: 0x1D
Reset: 0x03
Property: -

The register VERSION_NUM can be used for the radio transceiver identification and includes the version number of the device.

Bit	7	6	5	4	3	2	1	0
	VERSION_NUM[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	1

Bits 7:0 – VERSION_NUM[7:0] VERSION_NUM

Table 14-107. VERSION_NUM

Value	Description
0x03	Revision C

14.8.29 MAN_ID_0

Name: MAN_ID_0
Offset: 0x1E
Reset: 0x1F
Property: -

Part one of the JEDEC manufacturer ID.

Bit	7	6	5	4	3	2	1	0
	MAN_ID_0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	1	1	1	1

Bits 7:0 – MAN_ID_0[7:0] MAN_ID_0

Table 14-108. MAN_ID_0

Value	Description
0x1F	JEDEC manufacturer ID, bits[7:0] of the 32-bit JEDEC manufacturer ID are stored in register bits MAN_ID_0. Bits [15:8] are stored in register 0x1F (MAN_ID_1). The higher 16 bits of the ID are not stored in registers.

14.8.30 MAN_ID_1

Name: MAN_ID_1
Offset: 0x1F
Reset: 0x00
Property: -

Part two of the JEDEC manufacturer ID.

Bit	7	6	5	4	3	2	1	0
	MAN_ID_1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – MAN_ID_1[7:0] MAN_ID_1

Table 14-109. MAN_ID_1

Value	Description
0x00	JEDEC manufacturer ID, bits[15:8] of the 32-bit JEDEC manufacturer ID are stored in register bits MAN_ID_1. Bits [7:0] are stored in register 0x1E (MAN_ID_0). The higher 16 bits of the ID are not stored in registers.

14.8.31 SHORT_ADDR_0

Name: SHORT_ADDR_0
Offset: 0x20
Reset: 0xFF
Property: -

This register contains the lower 8-bit of the MAC short address for Frame Filter address recognition, bits[7:0].

Bit	7	6	5	4	3	2	1	0
	SHORT_ADDR_0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – SHORT_ADDR_0[7:0] SHORT_ADDR_0

14.8.32 SHORT_ADDR_1

Name: SHORT_ADDR_1
Offset: 0x21
Reset: 0xFF
Property: -

This register contains the higher 8-bit of the MAC short address for Frame Filter address recognition, bits[15:8].

Bit	7	6	5	4	3	2	1	0
	SHORT_ADDR_1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – SHORT_ADDR_1[15:8] SHORT_ADDR_1

14.8.33 PAN_ID_0

Name: PAN_ID_0
Offset: 0x22
Reset: 0xFF
Property: -

This register contains the lower 8-bit of the MAC PAN ID for Frame Filter address recognition, bits[7:0].

Bit	7	6	5	4	3	2	1	0
	PAN_ID_0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – PAN_ID_0[7:0] PAN_ID_0

14.8.34 PAN_ID_1

Name: PAN_ID_1
Offset: 0x23
Reset: 0xFF
Property: -

This register contains the higher 8-bit of the MAC PAN ID for Frame Filter address recognition, bits[15:8].

Bit	7	6	5	4	3	2	1	0
	PAN_ID_1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 7:0 – PAN_ID_1[7:0] PAN_ID_1

14.8.35 IEEE_ADDR

Name: IEEE_ADDR
Offset: 0x24 + n*0x01 [n=0..7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	IEEE_ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IEEE_ADDR[7:0] IEEE_ADDR
The IEEE_ADDR_[0..7] registers contains the MAC IEEE address for Frame Filter address recognition.
IEEE_ADDR_0: bits[7:0].
IEEE_ADDR_1: bits[15:8]
IEEE_ADDR_2: bits[23:16]
IEEE_ADDR_3: bits[31:24]
IEEE_ADDR_4: bits[39:32]
IEEE_ADDR_5: bits[47:40]
IEEE_ADDR_6: bits[55:48]
IEEE_ADDR_7: bits[63:56]

14.8.36 XAH_CTRL_0

Name: XAH_CTRL_0
Offset: 0x2C
Reset: 0x38
Property: -

The XAH_CTRL_0 register is a control register for Extended Operating Mode.

Bit	7	6	5	4	3	2	1	0
	MAX_FRAME_RETRIES[3:0]				MAX_CSMA_RETRIES[2:0]			SLOTTED_OPERATION
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	1	0	0	0

Bits 7:4 – MAX_FRAME_RETRIES[3:0] MAX_FRAME_RETRIES
Number of retransmission attempts in TX_ARET mode before the transaction gets cancelled.

Table 14-110. MAX_FRAME_RETRIES

Value	Description
0x3	The setting of MAX_FRAME_RETRIES in TX_ARET mode specifies the number of attempts to retransmit a frame, when it was not acknowledged by the recipient, before the transaction gets canceled. Valid values are [0x7, 0x6, …, 0x0].

Bits 3:1 – MAX_CSMA_RETRIES[2:0] MAX_CSMA_RETRIES
Number of retries in TX_ARET mode to repeat the CSMA-CA procedure before the transaction gets cancelled.

Table 14-111. MAX_CSMA_RETRIES

Value	Description
0x0(1)	No retries
0x1(1)	One retry
0x2(1)	Two retries
0x3(1)	Three retries
0x4(1)	Four retries
0x5(1)	Five retries
0x6	Reserved
0x7(2)	Immediate frame transmission without performing CSMA-CA

1. MAX_CSMA_RETRIES specifies the number of retries in TX_ARET mode to repeat the CSMA-CA procedure before the transaction gets cancelled. According to IEEE 802.15.4 the valid range of MAX_CSMA_RETRIES is [5, 4, …, 0].
2. A value of MAX_CSMA_RETRIES = 7 initiates an immediate frame transmission without performing CSMACA. No retry is performed. This may especially be required for slotted acknowledgment operation.

Bit 0 – SLOTTED_OPERATION SLOTTED_OPERATION
For RX_AACK mode, the register bit SLOTTED_OPERATION determines, if the transceiver will require a time base for slotted operation.

Table 14-112. SLOTTED_OPERATION

Value	Description
0x0	The radio transceiver operates in unslotted mode. An acknowledgment frame is automatically sent if requested.
0x1	The transmission of an acknowledgment frame has to be controlled by the microcontroller.

Using RX_AACK mode in networks operating in beacon or slotted mode, refer to IEEE 802.15.4-2006, Section 5.5.1, register bit SLOTTED_OPERATION indicates that acknowledgement frames are to be sent on backoff slot boundaries (slotted acknowledgement).
If this register bit is set the acknowledgement frame transmission has to be initiated by the microcontroller using the rising edge of pin 11 (SLP_TR). This waiting state is signaled in register bits TRAC_STATUS (register 0x02, TRX_STATE) with value SUCCESS_WAIT_FOR_ACK.

14.8.37 CSMA_SEED_0

Name: CSMA_SEED_0
Offset: 0x2D
Reset: 0xEA
Property: -

The register CSMA_SEED_0 contains the lower 8-bit of CSMA_SEED.

Bit	7	6	5	4	3	2	1	0
	CSMA_SEED_0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	0	1	0

Bits 7:0 – CSMA_SEED_0[7:0] CSMA_SEED_0
Lower 8-bit of CSMA_SEED, bits [7:0]. Used as seed for random number generation in the CSMA-CA algorithm.

Table 14-113. CSMA_SEED_0

Value	Description
0xEA	This register contains the lower 8-bit of the CSMA_SEED, bits [7:0]. The higher 3-bit are part of register bits CSMA_SEED_1 (register 0x2E, CSMA_SEED_1). CSMA_SEED is the seed for the random number generation that determines the length of the backoff period in the CSMA-CA algorithm.

1. It is recommended to initialize register bits CSMA_SEED_0 and CSMA_SEED_1 with random values. This can be done using register bits RND_VALUE (register 0x06, PHY_RSSI).
2. The content of register bits CSMA_SEED_0 and CSMA_SEED_1 initializes the TX_ARET random backoff generator after wakeup from DEEP_SLEEP state. It is recommended to re-initialize both registers after every DEEP_SLEEP state with a random value.

14.8.38 CSMA_SEED_1

Name: CSMA_SEED_1
Offset: 0x2E
Reset: 0x42
Property: -

The CSMA_SEED_1 register is a control register for RX_AACK and contains a part of the CSMA_SEED for the CSMACA algorithm.

Bit	7	6	5	4	3	2	1	0
	AACK_FVN_MODE[1:0]		AACK_SET_PD	AACK_DIS_ACK	AACK_I_AM_COORD	CSMA_SEED_1[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	1	0

Bits 7:6 – AACK_FVN_MODE[1:0] AACK_FVN_MODE
The register bits AACK_FVN_MODE control the ACK behavior dependent on FCF frame version number within RX_AACK mode.

Table 14-114. AACK_FVN_MODE

Value	Description
0x0	Accept frames with version number 0
0x1	Accept frames with version number 0 or 1
0x2	Accept frames with version number 0 or 1 or 2
0x3	Accept frames independent of frame version number

1. AACK_FVN_MODE value one indicates frames according to [2] IEEE 802.15.4–2006, a value of three indicates frames according to [1] IEEE 802.15.4–2003 standard.

The frame control field of the MAC header (MHR) contains a frame version subfield. The setting of register bits AACK_FVN_MODE specifies the frame filtering behavior of the AT86RF212B. According to the content of these register bits the radio transceiver passes frames with a specific frame version number, number group, or independent of the frame version number.
Thus the register bits AACK_FVN_MODE defines the maximum acceptable frame version. Received frames with a higher frame version number than configured do not pass the frame filter and are not acknowledged.
The frame version field of the acknowledgment frame is set to zero according to [2] IEEE 802.15.4-2006, Section 7.2.2.3.1 Acknowledgment frame MHR fields.

Bit 5 – AACK_SET_PD AACK_SET_PD
The content of AACK_SET_PD bit is copied into the frame pending subfield of the acknowledgment frame if the ACK is the response to a data request MAC command frame.

Table 14-115. AACK_SET_PD

Value	Description
0x0	Pending data bit set to zero
0x1	Pending data bit set to one

In addition, if register bits AACK_FVN_MODE (register 0x2E, CSMA_SEED_1) are configured to accept frames with a frame version other than zero or one, the content of register bit AACK_SET_PD is also copied into the frame pending subfield of the acknowledgment frame for any MAC command frame with a frame version of two or three that have the security enabled subfield set to one. This is done with the assumption that a future version of the [2] IEEE 802.15.4-2006 standard might change the length or structure of the auxiliary security header.

Bit 4 – AACK_DIS_ACK AACK_DIS_ACK
If this bit is set no acknowledgment frames are transmitted in RX_AACK Extended Operating Mode, even if requested.

Table 14-116. AACK_DIS_ACK

Value	Description
0x0	Acknowledgment frames are transmitted
0x1	Acknowledgment frames are not transmitted

Bit 3 – AACK_I_AM_COORD AACK_I_AM_COORD
This register bit has to be set if the node is a PAN coordinator. It is used for frame filtering in RX_AACK.

Table 14-117. AACK_I_AM_COORD

Value	Description
0x0	PAN coordinator addressing is disabled
0x1	PAN coordinator addressing is enabled

If AACK_I_AM_COORD = 1 and if only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is the PAN coordinator and the source PAN identifier matches mac- *PANId*, for details refer to [2] IEEE 802.15.4-2006, Section 7.5.6.2 (third-level filter rule six).

Bits 2:0 – CSMA_SEED_1[2:0] CSMA_SEED_1
Higher 3-bit of CSMA_SEED, bits [10:8]. Seed for random number generation in the CSMA-CA algorithm.

Table 14-118. CSMA_SEED_1

Value	Description
0x2	These register bits are the higher 3-bit of the CSMA_SEED, bits [10:8]. The lower part is in register 0x2D (CSMA_SEED_0), see register CSMA_SEED_0 for details.

14.8.39 CSMA_BE

Name: CSMA_BE
Offset: 0x2F
Reset: 0x53
Property: -

The register CSMA_BE contains the backoff exponents for the CSMA-CA algorithm.

Note: If MIN_BE = 0 and MAX_BE = 0 the CCA backoff period is always set to zero.

Bit	7	6	5	4	3	2	1	0
	MAX_BE[3:0]				MIN_BE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	1	1

Bits 7:4 – MAX_BE[3:0] MAX_BE
Maximum backoff exponent in the CSMA-CA algorithm.

Table 14-119. MAX_BE

Value	Description
0x5	Register bits MAX_BE defines the maximum backoff exponent used in the CSMA-CA algorithm to generate a pseudo random number for CCA backoff. Valid values are [0x8, 0x7, …, 0x0].

For details refer to [2] IEEE 802.15.4-2006, Section 7.5.1.4.

Bits 3:0 – MIN_BE[3:0] MIN_BE
Minimum backoff exponent in the CSMA-CA algorithm.

Table 14-120. MIN_BE

Value	Description
0x3	Register bits MIN_BE defines the minimum backoff exponent used in the CSMA-CA algorithm to generate a pseudo random number for CCA backoff. Valid values are [MAX_BE, (MAX_BE – 1), …, 0x0].

14.9 Reset Values

The reset values of the AT86RF212B registers in state P_ON[1, 2, 3] are shown in the table below.

Note: All reset values in the table below are only valid after a power on reset. After a reset procedure (/RST = L), the reset values of selected registers (for example registers 0x01, 0x10, 0x11, 0x30) can differ from the reset values listed in the table.

Address	Reset Value
0x00	0x00
0x01	0x00
0x02	0x00
0x03	0x19
0x04	0x20
0x05	0x60
0x06	0x00

.....continued

Address	Reset Value
0x07	0xFF
0x08	0x25
0x09	0x77
0x0A	0x17
0x0B	0xA7
0x0C	0x24
0x0D	0x01
0x0E	0x00
0x0F	0x00

Address	Reset Value
0x10	0x00
0x11	0x02
0x12	0xF0
0x13	0x00
0x14	0x00
0x15	0x00
0x16	0x31
0x17	0x00
0x18	0x58
0x19	0x00
0x1A	0x48
0x1B	0x40
0x1C	0x07
0x1D	0x03
0x1E	0x1F
0x1F	0x00

Address	Reset Value
0x20	0xFF
0x21	0xFF
0x22	0xFF
0x23	0xFF
0x24	0x00
0x25	0x00
0x26	0x00

.....continued

Address	Reset Value
0x27	0x00
0x28	0x00
0x29	0x00
0x2A	0x00
0x2B	0x00
0x2C	0x38
0x2D	0xEA
0x2E	0x42
0x2F	0x53

Address	Reset Value
0x30	0x00
0x31	0x00
0x32	0x00
0x33	0x00
0x34	0x3F
0x35	0x00
0x36	0x00
0x37	0x00
0x38	0x00
0x39	0x40
0x3A	0x00
0x3B	0x00
0x3C	0x00
0x3D	0x00
0x3E	0x00
0x3F	0x00

Notes:

1. While the reset value of register 0x10 is 0x00, any practical access to the register is only possible when DVREG is active. So this register is always read out as 0x04.
2. While the reset value of register 0x11 is 0x02, any practical access to the register is only possible when BATMON is activated. So this register is always read out as 0x22 in P_ON state.
3. While the reset value of register 0x30 is 0x00, any practical access to the register is only possible when the radio transceiver is accessible. So the register is usually read out as:
   - 3.1. 0x11 after a reset in P_ON state
   - 3.2. 0x07 after a reset in any other state

Related Links

14.2.1.4.5 Reset Procedure

15. Electrical Characteristics

15.1 Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

15.2 Absolute Maximum Ratings

Stresses beyond those listed in Table 15-1 may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 15-1. Absolute Maximum Ratings

Symbol	Description	Min.	Max.	Units
V_{DD}	Power supply voltage	0	3.8	V
I_{VDD}	Current into a V_{DD} pin	-	46(1)	mA
I_{GND}	Current out of a GND pin	-	65(1)	mA
V_{PIN}	Pin voltage with respect to GND and V_{DD} $V_{DD\ MAX}$=3.6V	GND-0.6V	VDD+0.6V	V
V_{ESD}	ESD robustness	Human Body Model (HBM)[8] Charged Device Model (CDM)[9]	4 450	kV V
P_{RF}	Input RF level		+10	dB
$T_{storage}$	Storage temperature	-50	150	°C
T_{LEAD}	T=10s (soldering profile compliant with IPC/JEDEC J STD 020B)	-	260	°C

⚠ CAUTION This device is sensitive to electrostatic discharges (ESD). Improper handling may lead to permanent performance degradation or malfunctioning.
Handle the device following best practice ESD protection rules: Be aware that the human body can accumulate charges large enough to impair functionality or destroy the device.

Related Links

23. References

15.3 General Operating Ratings

The device must operate within the ratings listed in Table 15-2 in order for all other electrical characteristics and typical characteristics of the device to be valid.

Table 15-2. General Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDIN}	Power supply voltage	1.8	3.3	3.63	V
V_{DDIO}	IO Supply Voltage	1.8	3.3	3.63	V
V_{DDANA}	Analog supply voltage	1.8	3.3	3.63	V
T_A	Temperature range	-40	25	85	°C
T_J	Junction temperature	-	-	100	°C

⚠ CAUTION In debugger cold-plugging mode, NVM erase operations are not protected by the BOD33 and BOD. NVM erase operation at supply voltages below specified minimum can cause corruption of NVM areas that are mandatory for correct device behavior.

15.4 Injection Current

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 15-3. Injection Current(1,2)

Symbol	Description	min	max	Unit
I_{inj1} (3)	IO pin injection current	-1	+1	mA
I_{inj2} (4)	IO pin injection current	-15	+15	mA
$I_{injtotal}$	Sum of IO pins injection current	-45	+45	mA

Notes:

1. Injecting current may have an effect on the accuracy of Analog blocks
2. Injecting current on Backup IOs is not allowed
3. Conditions for V_{pin}: V_{pin} < GND-0.6V or 3.6V<V_{pin}≤4.2V.

   Conditions for V_{DD}: 3V<V_{DD}≤3.6V.

   If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor *R* is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj1}|$.

   If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor *R* is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj1}$.
4. Conditions for V_{pin}: V_{pin} < GND-0.6V or V_{pin}≤3.6V.

   Conditions for V_{DD}: V_{DD}≤3V.

   If V_{pin} is lower than GND-0.6V, a current limiting resistor is required. The negative DC injection current limiting resistor *R* is calculated as $R = |(GND-0.6V - V_{pin})/I_{inj2}|$.

   If V_{pin} is greater than V_{DD}+0.6V, a current limiting resistor is required. The positive DC injection current limiting resistor *R* is calculated as $R = (V_{pin}-(V_{DD}+0.6))/I_{inj2}$.

15.5 Supply Characteristics

Table 15-4. Supply Characteristics

Symbol	Voltage		
	Min.	Max.	Units
V_{DDIO}	1.8	3.63	V
V_{DDIN}	1.8	3.63	V
V_{DDANA}	1.8	3.63	V
V_{BAT}	1.8	3.63	V

Table 15-5. Supply Slew Rates(1)

Symbol	Fall Rate	Rise Rate	Units
	Max.	Max.	
V_{DDIO}	0.05	0.1	V/µs
V_{DDIN}	0.05	0.1	V/µs
V_{DDANA}	0.05	0.1	V/µs
V_{BAT}	0.05	0.1	V/µs

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Related Links

8. Power Supply and Start-Up Considerations

15.6 Maximum Clock Frequencies

Table 15-6. Maximum GCLK Generator Output Frequencies(1)

Symbol	Description	Conditions	Fmax		Units
			PL0	PL2	
$F_{gclkgen}$[0:2]	GCLK Generator output Frequency	-	24	96	MHz
$F_{gclkgen}$[3:8]		undivided	24	96	MHz
$F_{gclkgen}$[3:8]		divided	16	66	MHz

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 15-7. Maximum Peripheral Clock Frequencies(1)

Symbol	Description	Conditions	Max.		Units
			PL0	PL2	
f_{CPU}	CPU clock frequency	-	12	48	MHz
f_{AHB}	AHB clock frequency	-	12	48	MHz
f_{APBA}	APBA clock frequency	Bus clock domain = BACKUP	6	6	MHz
f_{APBA}	APBA clock frequency	Bus clock domain = Low Power	12	48	MHz

.....continued

Symbol	Description	Conditions	Max.		Units
			PL0	PL2	
f_{APBB}	APBB clock frequency	-	12	48	MHz
f_{APBC}	APBC clock frequency	-			
f_{APBD}	APBD clock frequency	-			
f_{APBE}	APBE clock frequency	-			
$f_{GCLK\_DFLL48M\_REF}$	DFLL48M Reference clock frequency	-	NA	33	kHz
f_{GCLK_DPLL}	FDPLL96M Reference clock frequency	-	2	2	MHz
$f_{GCLK\_DPLL\_32K}$	FDPLL96M 32k Reference clock frequency	-	32	100	kHz
f_{GCLK_EIC}	EIC input clock frequency	-	12	48	MHz
f_{GCLK_USB}	USB input clock frequency	-	NA	60	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_0}$	EVSYS channel 0 input clock frequency	-	12	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_1}$	EVSYS channel 1 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_2}$	EVSYS channel 2 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_3}$	EVSYS channel 3 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_4}$	EVSYS channel 4 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_5}$	EVSYS channel 5 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_6}$	EVSYS channel 6 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_7}$	EVSYS channel 7 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_8}$	EVSYS channel 8 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_9}$	EVSYS channel 9 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_10}$	EVSYS channel 10 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_11}$	EVSYS channel 11 input clock frequency	-			
$f_{GCLK\_SERCOMx\_SLOW}$	Common SERCOM slow input clock frequency	-	1	5	MHz
$f_{GCLK\_SERCOM0\_CORE}$	SERCOM0 input clock frequency	-	12	48	MHz
$f_{GCLK\_SERCOM1\_CORE}$	SERCOM1 input clock frequency	-			
$f_{GCLK\_SERCOM2\_CORE}$	SERCOM2 input clock frequency	-			
$f_{GCLK\_SERCOM3\_CORE}$	SERCOM3 input clock frequency	-			
$f_{GCLK\_SERCOM4\_CORE}$	SERCOM4 input clock frequency	-			
$f_{GCLK\_SERCOM5\_CORE}$	SERCOM5 input clock frequency	-			
$f_{GCLK\_TCC0,\ GCLK\_TCC1}$	TCC0,TCC1 input clock frequency	-	24	96	MHz
$f_{GCLK\_TCC2,\ GCLK\_TC0}$	TCC2,TC0 input clock frequency	-	12	48	MHz
f_{GCLK_TC1}	TC1 input clock frequency	-			
f_{GCLK_TC4}	TC4 input clock frequeny	-			

.....continued					
Symbol	Description	Conditions	Max.		Units
			PL0	PL2	
f_{GCLK_ADC}	ADC input clock frequency	-	12	48	MHz
f_{GCLK_AC}	AC digital input clock frequency	-			
f_{GCLK_PTC}	PTC input clock frequency	-			
f_{GCLK_CCL}	CCL input clock frequency	-			

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

15.7 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - V_{DDIN} = 3.3V or 1.8V
  - For V_{DDIN}=1.8V, CPU is running on Flash with 3 wait states
  - For V_{DDIN}=3.3V, CPU is running on Flash with 1 wait state in PL0 and 2 wait states in PL2.
  - Low power cache is enabled
  - BOD33 is disabled
  - AT86RF212B is in sleep mode
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32KHz crystal oscillator) running with external 32KHz crystal
  - When in active Performance Level 2 (PL2) mode, DFLL48M is running at 48MHz and using XOSC32K as reference
  - When in active PL0 mode, the internal Multi RC Oscillator is running at specified frequency
- **Clocks**
  - DFLL48M used as main clock source when in PL2 mode. In PL0 mode, OSC16M used at 4, 8, or 12MHz
  - Clock masks and dividers at reset values: All AHB & APB clocks enabled, CPUDIV=1, BUPDIV=1, LPDIV=1
  - I/Os are configured in Digital Functionality Disabled mode

Table 15-8. Active Current Consumption(1)

Mode	Conditions	Regulator	PL	Clock	Vcc	Ta	Typ.	Max.	Units
ACTIVE	COREMARK	LDO	PL0	OSC 12MHz	1.8V	Max at 85°C Typ at 25°C	77	106	µA/MHz
					3.3V		79	101	
				OSC 8MHz	1.8V		80	111	
					3.3V		82	120	
				OSC 4MHz	1.8V		89	153	
					3.3V		92	166	
			PL2	DFLL 48MHz	1.8V		93	103	
					3.3V		95	105	
	FIBO	LDO	PL0	OSC 12MHz	1.8V		76	95	
					3.3V		78	98	
				OSC 8MHz	1.8V		79	111	
					3.3V		81	119	
				OSC 4MHz	1.8V		89	155	
					3.3V		91	173	
			PL2	DFLL 48MHz	1.8V		94	104	
					3.3V		95	104	
ACTIVE	WHILE1	LDO	PL0	OSC 12MHz	1.8V	Max at 85°C Typ at 25°C	60	80	µA/MHz
					3.3V		62	87	
				OSC 8MHz	1.8V		63	95	
					3.3V		65	106	
				OSC 4MHz	1.8V		72	138	
					3.3V		75	153	
			PL2	DFLL 48MHz	1.8V		73	80	
					3.3V		73	81	
IDLE			PL0	OSC 12MHz	1.8V		17	30	
					3.3V		13	24	

Note: 1. These values are based on characterization.

Table 15-9. Standby and Backup Mode Current Consumption(1)

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
STANDBY	PD0, PD1, PD2 in retention state	LPEFF Disable	1.8V	25°C	1.5	3.44	µA
				85°C	24.9	64.2	
		LPEFF Enable	3.3V	25°C	1.4	3.04	
				85°C	21.7	49.2	
	PD0, PD1, PD2 in retention state with RTC running on OSCULP32K	LPEFF Disable	1.8V	25°C	1.7	3.54	
				85°C	25.0	64.2	
		LPEFF Enable	3.3V	25°C	1.6	3.24	
				85°C	21.8	49.3	
	PD1, PD2 in retention state and PD0 in active state	LPEFF Disable	1.8V	25°C	1.8	3.94	
				85°C	27.3	76.3	
		LPEFF Enable	3.3V	25°C	1.6	3.54	
				85°C	24.4	56.3	
	PD2 in retention state and PD0, PD1 in active state	LPEFF Disable	1.8V	25°C	2.5	5.94	
				85°C	51.7	129.6	
		LPEFF Enable	3.3V	25°C	2.2	5.14	
				85°C	31.7	85.4	
	PD0, PD1, PD2 in active state	LPEFF Disable	1.8V	25°C	3.5	11.3	
				85°C	81.9	188.9	
		LPEFF Enable	3.3V	25°C	3.0	6.64	
				85°C	51.4	151.1	

.....continued

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
BACKUP	powered by VDDIN VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.68	1.17	µA
				85°C	5.4	13.6	
			3.3V	25°C	0.61	1.44	
				85°C	1.9	17.4	
	powered by VDDIN VBAT consumption		1.8V	25°C	0.201	0.344	
				85°C	1.011	3.427	
			3.3V	25°C	0.207	0.070	
				85°C	1.036	3.476	
	powered by VDDIN with RTC running on OSCULP32K VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.73	1.23	
				85°C	5.5	13.6	
			3.3V	25°C	0.84	1.54	
				85°C	1.9	17.5	
	powered by VDDIN with RTC running on OSCULP32K VBAT consumption		1.8V	25°C	0.201	0.345	
				85°C	1.011	3.428	
			3.3V	25°C	0.208	0.358	
				85°C	1.027	3.465	
	powered by VBAT VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.11	0.25	
				85°C	2.4	5.3	
			3.3V	25°C	0.19	0.5	
				85°C	3.8	9.0	
	powered by VBAT VBAT consumption		1.8V	25°C	0.37	0.60	
				85°C	2.0	4.9	
			3.3V	25°C	0.40	0.64	
				85°C	2.1	5.1	
	powered by VBAT with RTC running on OSCULP32K VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.14	0.25	
				85°C	2.4	5.3	
			3.3V	25°C	0.20	0.5	
				85°C	3.8	9.0	
	powered by VBAT with RTC running on OSCULP32K VBAT consumption		1.8V	25°C	0.42	0.66	
				85°C	2.1	5.0	
			3.3V	25°C	0.45	0.70	
				85°C	2.2	5.2	

Note: 1. These values are based on characterization.

15.8 Wake-Up Time

Conditions:

- V_{DDIN} = 3.3V
- LDO Regulation mode
- CPU clock = OSC16M at 12 MHz
- 1 Wait-state
- Cache enabled
- Flash Fast Wake up enabled (NVMCTRL.CTRLB.FWUP = 1)
- Flash in WAKEUPINSTANT mode (NVMCTRL.CTRLB.SLEEPPRM = 1)

For IDLE and STANDBY, CPU sets an IO by writing PORT->IOBUS without jumping in an interrupt handler (Cortex M0+ register PRIMASK = 1). The wake-up time is measured between the edge of the wake-up input signal and the edge of the GPIO pin.

For Backup, the exit of mode is done through RTC wake-up. The wake-up time is measured between the toggle of the RTC pin and the set of the IO done by the first executed instructions after reset.

For OFF mode, the exit of mode is done through the reset pin, the time is measured between the rising edge of the RESETN signal and the set of the IO done by the first executed instructions after reset.

Table 15-10. Wake-Up Timing

Sleep Mode	Condition		Typ.	Unit
IDLE	PL2 or PL0		1.2	µs
STANDBY	PL0 PM.PLSEL.PLDIS = 1 (see errata 13674 for revision A)	PDCFG default	5.1	µs
		PD012 forced active PDCFG = 3	2.1	
	PL2 Voltage scaling at default values: SUPC->VREG.VSVSTEP = 0 SUPC->VREG.VSPER = 0	PDCFG default	76	µs
		PD012 forced active PDCFG = 3	75	
	PL2 Voltage scaling at fastest setting: SUPC->VREG.VSVSTEP = 15 SUPC->VREG.VSPER = 0	PDCFG default	16	µs
		PD012 forced active PDCFG = 3	15	µs
BACKUP	—		90	µs

15.9 I/O Pin Characteristics

Note: The I/O pin characteristics numbers are for the micronontroller SAM L21.

There are three different pin types with three different speeds: Backup, Normal, and High Sink(3,4).

The Drive Strength bit is located in the Pin Configuration register of the PORT (PORT.PINCFG.DRVSTR).

Table 15-11. I/O Pins Common Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
V_{IL}	Input low-level voltage	V_{DD}=1.8V-2.7V	-	-	0.25*V_{DD}	V
		V_{DD}=2.7V-3.63V	-	-	0.3*V_{DD}	
V_{IH}	Input high-level voltage	V_{DD}=1.8V-2.7V	0.7*V_{DD}	-	-	
		V_{DD}=2.7V-3.63V	0.55*V_{DD}	-	-	
V_{OL}	Output low-level voltage	V_{DD}>1.8V, I_{OL} max	-	0.1*V_{DD}	0.2*V_{DD}	
V_{OH}	Output high-level voltage	V_{DD}>1.8V, I_{OH} max	0.8*V_{DD}	0.9*V_{DD}	-	
R_{PULL}	Pull-up - Pull-down resistance	All pins except PA24, PA25	20	40	60	kΩ
		PA24, PA25(1)	50	100	150	
I_{LEAK}	Input leakage current	Pull-up resistors disabled	-1	+/-0.015	1	µA

Table 15-12. I/O Pins Maximum Output Current

Symbol	Parameter	Conditions	Backup Pins in Backup Mode(4)	Backup and Normal Pins(4)	High Sink Pins(3)	Backup and Normal Pins(4)	High Sink Pins(3)	Units
				DRVSTR=0		DRVSTR=1		
I_{OL}	Maximum Output low-level current	V_{DD}=1.8V-3V	0.005	1	2	2	4	mA
		V_{DD}=3V-3.63V	0.008	2.5	6	6	12	
I_{OH}	Maximum Output high-level current	V_{DD}=1.8V-3V	0.005	0.7	1.5	1.5	3	
		V_{DD}=3V-3.63V	0.008	2	5	5	10	

Table 15-13. I/O Pins Dynamic Characteristics(1)

Symbol	Parameter	Conditions	Backup Pins in Backup Mode(4)	Backup and Normal Pins(4)	High Sink Pins(3)	Backup and Normal Pins(4)	High Sink Pins(3)	Units
				DRVSTR=0		DRVSTR=1		
t_{RISE}	Maximum Rise time	V_{DD}=3.3V, load=20pF	2000	13	6	6	4.5	ns
t_{FALL}	Maximum Fall time	V_{DD}=3.3V, load=20pF	2000	12	7	7	4.5	

The pins with I^2C alternative mode available are compliant(2) with I^2C norms. All I^2C pins support Standard mode (Sm), Fast mode (Fm), Fast plus mode (Fm+), and High speed mode (Hs, up to 3.4MHz). The available I^2C pins are listed in the I/O Multiplexing section.

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. The pins PA12, PA13, PB12, PB13, PB16, PB17, PB30, PB31 are limited on output low-level current in I^2C Standard mode (*Sm*) and Fast mode (*Fm*). The limitation is 2.5mA instead of 3mA for V_{OL}=0.4V, and 3mA instead of 6mA for V_{OL}=0.6V.
3. The following pins are High Sink pins and have different properties than normal pins: PA08, PA09, PA16, PA17, PA22, PA23, PA27, PA31.
4. The following pins are Backup pins and have different properties than normal pins: PA00, PA01, PB00, PB01, PB02, PB03.

Related Links
7.3.3 SERCOM I2C Pins

15.10 Analog Characteristics

15.10.1 Voltage Regulator Characteristics

15.10.1.1 LDO Regulator

Table 15-14. LDO Regulator Electrical Characteristics

Symbol	Parameter	Conditions	Typ.	Units
VREGSCAL	Voltage scaling	min step size for PLx to Ply transistion	5	mV
		Voltage Scaling Period(1)	1	µs

Note: 1. These are based on simulation. These values are not covered by test or characterization

Table 15-15. External Components Requirements in Linear Mode

Symbol	Parameter	Conditions	Typ.	Units
C_{IN}	Input regulator capacitor		4.7	µF
		Ceramic dielectric X7R	100	nF
C_{OUT}	Output regulator capacitor		1	µF
		Ceramic dielectric X7R	100	nF

15.10.2 APWS

Table 15-16. Automatic Power Switch Characteristics

Symbol	Parameters	Min.	Typ.	Max.	Unit
CD	Decoupling capacitor on VDDIN	-	4.7	-	µF
T_{HUP}	V_{DDIN} threshold	-	1.84	-	V
T_{HDWN}		-	1.75	-	V
T_{HHYS}	V_{DDIN} hysteresis	-	90	-	mV

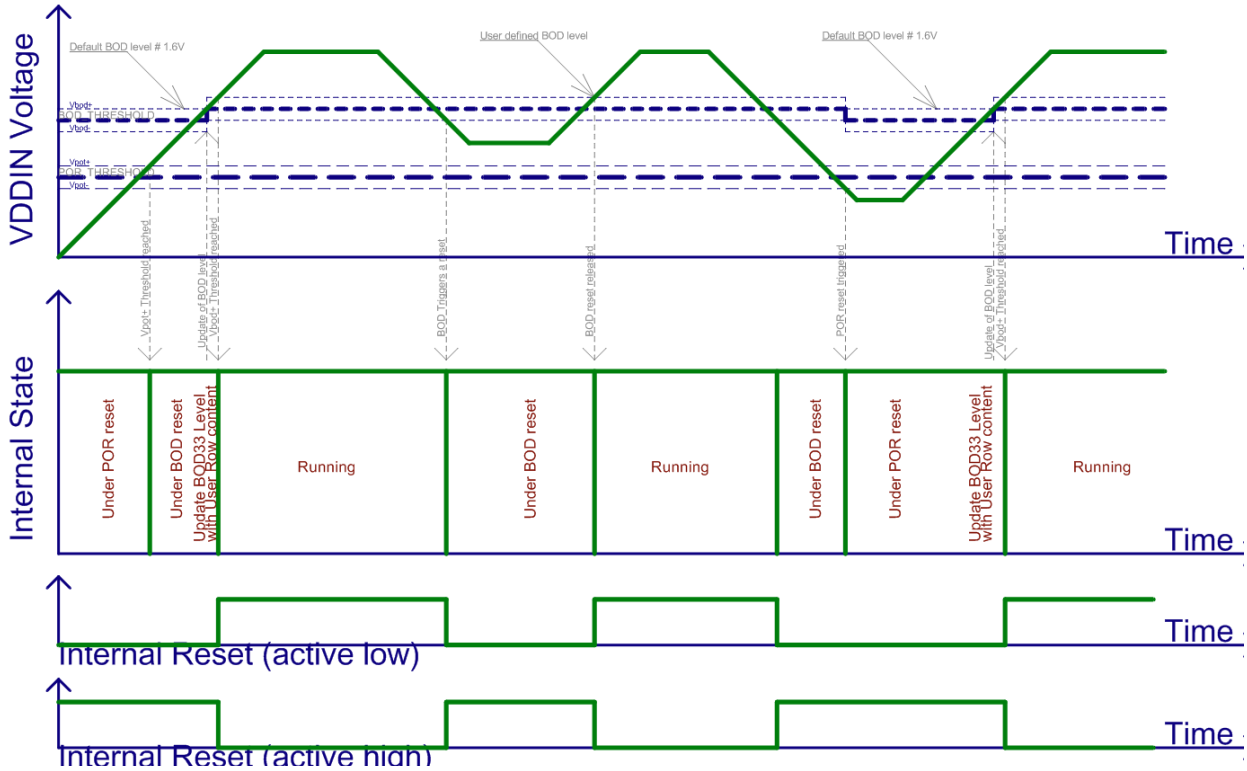
Note: With CDmin = Imax /(dV/dt)

15.10.3 Power-On Reset (POR) Characteristics

Table 15-17. POR33 Characteristics

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{POT+}	Voltage threshold Level on V_{DDIN} rising		1.53	1.58	1.62	V
V_{POT-}	Voltage threshold Level on V_{DDIN} falling		0.6	1.04	1.39	V

Figure 15-1. BOD Reset Behavior at Startup and Default Levels



15.10.4 Brown-Out Detectors (BOD) Characteristics

Table 15-18. BOD33 Characteristics(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{BOD+}	BOD33 high threshold Level	V_{BAT}, 15	1.67	1.74	1.81	V
		V_{DDIN}, 7	1.75	1.75	1.80	
		V_{DDIN}, 6	1.66	1.72	1.75	
		V_{BAT}, 55	2.80	2.90	3.01	
		V_{DDIN}, 39	2.65	2.87	2.95	
		V_{BAT}, 63	3.02	3.14	3.26	
		V_{DDIN}, 48	03.12	3.20	3.29	
V_{BOD-} / V_{BOD}	BOD33 low threshold Level	V_{BAT}, 15	1.60	1.66	1.72	V
		V_{DDIN}, 7	1.63	1.673	1.71	
		V_{DDIN}, 6	1.60	1.65	1.68	
		V_{BAT}, 55	2.70	2.81	2.92	
		V_{DDIN}, 39	2.70	2.77	2.84	
		V_{BAT}, 63	2.92	3.04	3.16	
		V_{DDIN}, 48	3.00	3.08	3.16	
	Step size	-		34		mV

.....continued						
Symbol	**Parameters**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
V_{HYS}	Hysteresis (V_{BOD+} - V_{BOD-}) BOD33.LEVEL= 0x0 to 0x3F	V_{BAT}	38	-	135	mV
		V_{DDIN}	47	-	155	
T_{START}	Startup time	time from enable to RDY	-	3.2	-	µs

Note:

1. These values are based on characterization.

Table 15-19. BOD33 Power Consumption(1)

Symbol	Conditions	V_{CC}	Ta	Min.	Typ.	Max.	Units
IDD	IDLE, mode CONT	1.8V	Max. at 85°C	-	17.9	21.5	µA
		3.3V		-	28.8	33.1	
	IDLE, mode SAMPL	1.8V	Typ. at 25°C	-	0.020	0.306	
		3.3V		-	0.033	0.197	
	STDBY, mode SAMPL	1.8V		-	0.087	0.231	
		3.3V		-	0.114	0.347	

Note:

1. These values are based on characterization.

Related Links

13.11.6.5.7 Hysteresis

15.10.5 Analog-to-Digital Converter (ADC) Characteristics

Table 15-20. Operating Conditions

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
RES	Resolution	-	-	-	12	bits
	Conversion speed	-	10	-	1000	ksps
fs	Sampling clock	-	10	-	1000	kHz
clk	ADC Clock frequency	-	-	fs.16	-	Hz
T_S	Sampling time	OFFCOMP=1	250	-	25000	ns
	Conversion range	Diff mode	$-V_{REF}$	-	V_{REF}	V
		Single-Ended mode	0	-	V_{REF}	
V_{REF}	Reference input	REFCOMP=1	1	-	V_{DDANA}-0.6	V
		REFCOMP=0	V_{DDANA}	-	V_{DDANA}	
V_{IN}	Input channel range	-	0	-	V_{DDANA}	V
V_{CMIN}	Input common mode voltage	For V_{REF} > 1.0V	0.7	-	V_{REF}-0.7	V
		For V_{REF}=1.0V	0.3	-	V_{REF}-0.3	

.....continued						
Symbol	**Parameters**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
CSAMPLE(1)	Input sampling capacitance	-	-	2.8	3.2	pF
RSAMPLE(1)	Input sampling on-resistance	-	-	-	1715	Ω
Rref(1)	Reference input source resistance	REFCOMP=1	-	-	5	kΩ

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 15-21. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
I_{DDANA}	Differential Mode	fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}$=1.6V	Max.85°C Typ.25°C	-	105	128	µA
		fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}$=3.6V		-	279	307	
	Differential Mode	fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=1.6V, V_{REF}=1.0V		-	175	231	µA
		fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=3.0V, V_{REF}=2.0V		-	300	374	
		fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=3.6V, V_{REF}=3.0V		-	356	438	
	Differential Mode	fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}$=1.6V		-	30	41	µA
		fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{CC}=V_{REF}$=3.6V		-	53	71	
	Differential Mode	fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=1.6V, V_{REF}=1.0V		-	95	139	µA
		fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=3.0V, V_{REF}=2.0V		-	115	178	
		fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=3.6V, V_{REF}=3.0V		-	122	187	

.....continued

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
I_{DDANA}	Single-Ended Mode	fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}=1.6V$	Max.85°C Typ.25°C	-	138	158	µA
		fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}=3.6V$		-	321	359	
	Single-Ended Mode	fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=1.6V$, $V_{REF}=1.0V$		-	203	257	µA
		fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=3.0V$, $V_{REF}=2.0V$		-	331	413	
		fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=3.6V$, $V_{REF}=3.0V$		-	388	482	
	Single-Ended Mode	fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{CC}=V_{REF}=1.6V$		-	46	62	µA
		fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}=3.6V$		-	89	120	
	Single-Ended Mode	fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{CC}=1.6V$, $V_{REF}=1.0V$		-	109	157	µA
		fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=3.0V$, $V_{REF}=2.0V$		-	138	211	
		fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=3.6V$, $V_{REF}=3.0V$		-	148	228	

Note: 1. These values are based on characterization.

Table 15-22. Differential Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
ENOB	Effective Number of bits	$V_{DDANA}=3.0V$ / $V_{ref}=2.0V$	9.6	10.5	10.6	bits
		$V_{DDANA}=1.6V/3.6V$ $V_{ref}=1.0V$	8.9	9.7	9.9	
		$V_{DDANA}=V_{ref}=1.6V$	10	10.5	11.1	
		$V_{DDANA}=V_{ref}=3.6V$	10.5	10.9	11.0	
TUE	Total Unadjusted Error	$V_{DDANA}=3.0V$, $V_{ref}=2.0V$	-	7.5	11	LSB
INL	Integral Non Linearity	$V_{DDANA}=3.0V$, $V_{ref}=2.0V$	-	+/-1.5	+/-2.1	LSB

.....continued

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+1.1/-1.0	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-0.7	+/-1.5	%
		External Reference voltage 3.0V	-	+/-0.2	+/-0.5	
		Reference bandgap voltage	-	+/-0.4	+/-4.4	
		VDDANA	-	+/-0.1	+/-0.4	
		VDDANA/2	-	+/-0.4	+/-1.3	
		VDDANA/1.6	-	+/-0.3	+/-0.9	
	Offset Error	External Reference voltage 1.0V	-	+/-1.1	+/-2.4	mV
		External Reference voltage 3.0V	-	+/-1.1	+/-3	
		Reference bandgap voltage	-	+/-2.3	+/-7.5	
		VDDANA	-	+/-0.9	+/-2.9	
		VDDANA/2	-	+/-1	+/-2.6	
		VDDANA/1.6	-	+/-1	+/-2.9	
SFDR	Spurious Free Dynamic Range	Fs=1MHz / Fin=13 kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	68	75	77	dB
SINAD	Signal to Noise and Distortion ratio		60	65	66	
SNR	Signal to Noise ratio		61	66	67	
THD	Total Harmonic Distortion		-74	-73	-67	
	Noise RMS	External Reference voltage	-	1.0	2.5	mV

Note: 1. These values are based on characterization.

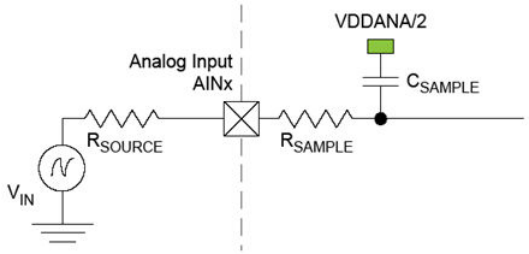
Table 15-23. Single-Ended Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
ENOB	Effective Number of bits	V_{DDANA}=3.0V / V_{ref} =2.0V	8.5	9.5	9.8	bits
		V_{DDANA}=1.6V/3.6V V_{ref}=1.0V	7.5	8.7	8.9	
		V_{DDANA}=V_{ref}=1.6V (2)	9.0	9.5	9.8	
		V_{DDANA}=V_{ref}=3.6V	9.2	9.8	9.9	
TUE	Total Unadjusted Error	V_{DDANA}=3.0V, V_{ref}=2.0V	-	17.4	31	LSB
INL	Integral Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-2.2	+/-10.1	LSB
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+/-0.9	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-1	+/-1.3	%
		External Reference voltage 3.0V	-	+/-0.3	+/-0.6	
		Reference bandgap voltage	-	+/-0.4	+/-3.2	
		V_{ref}=V_{DDANA} (2)	-	+/-0.1	+/-0.3	
		V_{ref}=V_{DDANA}/2	-	+/-0.6	+/-1.4	
		V_{ref}=V_{DDANA}/1.6	-	+/-0.4	+/-1	
	Offset Error	External Reference voltage 1.0V	-	+/-3.4	+/-13	mV
		External Reference voltage 3.0V	-	+/-3.6	+/-24	
		Reference bandgap voltage	-	+/-1	+/-14	
		V_{ref}=V_{DDANA} (2)	-	+/-4.2	+/-25	
		V_{ref}=V_{DDANA}/2	-	+/-5.7	+/-10	
		V_{ref}=V_{DDANA}/1.6	-	+/-6.3	+/-13	
SFDR	Spurious Free Dynamic Range	Fs=1MHz / Fin=13kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	65	71	78	dB
SINAD	Signal to Noise and Distortion ratio		53	59	61	
SNR	Signal to Noise ratio		53	59	61	
THD	Total Harmonic Distortion		-76	-70	-64	
	Noise RMS	External Reference voltage	-	2.0	7.0	mV

Notes:

1. These values are based on characterization.
2. All parameters given above exclude the corner V_{DDANA} = V_{ref} = 1.6V, Ta = -40°C.

Figure 15-2. ADC Analog Input AINx



The minimum sampling time $t_{samplehold}$ for a given R_{source} can be found using this formula:

$t_{\text{samplehold}} \geq \left(R_{\text{sample}} + R_{\text{source}}\right) \times C_{\text{sample}} \times \left(n + 2\right) \times \ln\left(2\right)$
For 12-bit accuracy:

$t_{\text{samplehold}} \geq \left(R_{\text{sample}} + R_{\text{source}}\right) \times C_{\text{sample}} \times 9.7$
where $t_{\text{samplehold}} \geq \frac{1}{2 \times f_{\text{ADC}}}$.

15.10.6 Analog Comparator (AC) Characteristics

Table 15-24. Electrical and Timing

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
	Positive and Negative input range voltage		0	-	V_{DDANA}	V
ICMR	Input common mode range		0	-	V_{DDANA}-0.1	V
Off(2)	Offset	COMPCTRLn.SPEED=0x0	-70	-4.5/+1.5	70	mV
		COMPCTRLn.SPEED=0x1	-55	-4.5/+1.5	55	
		COMPCTRLn.SPEED=0x2	-48	-4.5/+1.5	48	
		COMPCTRLn.SPEED=0x3	-42	-4.5/+1.5	42	
V_{Hys}(2)	Hysteresis	COMPCTRLn.HYST=0x0	10	45	74	mV
		COMPCTRLn.HYST=0x1	22	70	106	
		COMPCTRLn.HYST=0x2	37	90	116	
		COMPCTRLn.HYST=0x3	49	105	131	
T_{pd}(2)	Propagation Delay V_{cm}=V_{ddana}/2, V_{in}= +-100mV overdrive from V_{CM}	COMPCTRLn.SPEED=0x0	-	4	12.3	µs
		COMPCTRLn.SPEED=0x1	-	0.97	2.59	
		COMPCTRLn.SPEED=0x2	-	0.56	1.41	
		COMPCTRLn.SPEED=0x3	-	0.33	0.77	
T_{start}(1)	Start-up time	COMPCTRLn.SPEED=0x0	-	17	56	µs
		COMPCTRLn.SPEED=0x1	-	0.85	4.6	
		COMPCTRLn.SPEED=0x2	-	0.55	3.2	
		COMPCTRLn.SPEED=0x3	-	0.45	2.7	
V_{scale}(2)	INL			0.34		LSB
	DNL			0.06		
	Offset Error			0.1		
	Gain Error			1.22		

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 15-25. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ	Max.	Unit
I_{DDANA}	Current consumption $V_{CM}=V_{DDANA}/2$, +/-100mV overdrive from V_{CM}, voltage scaler disabled	COMPCTRLn.SPEED=0x0, V_{DDANA}=3.3V	Max.85°C Typ.25°C	-	50	1973	nA
		COMPCTRLn.SPEED=0x1, V_{DDANA}=3.3V		-	156	2082	
		COMPCTRLn.SPEED=0x2, V_{DDANA}=3.3V		-	289	2223	
		COMPCTRLn.SPEED=0x3, V_{DDANA}=3.3V		-	549	2495	
	Current consumption Voltage Scaler only	V_{DDANA}=3.3V		-	13	17	µA

Note:
1. These values are based on characterization.

15.10.7 DETREF

Table 15-26. Reference Voltage Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
ADC Ref	ADC internal reference	nom. 1.0V, V_{CC} = 3.0V, T = 25°C	0.967	1.0	1.017	V
		nom. 1.1V, V_{CC} = 3.0V, T = 25°C	1.069	1.1	1.120	
		nom. 1.2V, V_{CC} = 3.0V, T = 25°C	1.167	1.2	1.227	
		nom. 1.25V, V_{CC} = 3.0V, T = 25°C	1.214	1.25	1.280	
		nom. 2.0V, V_{CC} = 3.0V, T = 25°C	1.935	2.0	2.032	
		nom. 2.2V, V_{CC} = 3.0V, T = 25°C	2.134	2.2	2.242	
		nom. 2.4V, V_{CC} = 3.0V, T = 25°C	2.328	2.4	2.458	
		nom. 2.5V, V_{CC} = 3.0V, T = 25°C	2.420	2.5	2.565	
	Ref Temperature coefficient	drift over [-40, +25]°C	—	-0.01/+0.015	—	%/°C
		drift over [+25, +85]°C	—	-0.01/0.005	—	
	Ref Supply coefficient	drift over [1.6, 3.6]V	—	+/-0.35	—	%/V
AC Ref	AC Ref Accuracy	V_{CC} = 3.0V, T = 25°C	1.073	1.1	1.123	V
	Ref Temperature coefficient	drift over [-40, +25]°C	—	+/-0.01	—	%/°C
		drift over [+25, +85]°C	—	-0.005/+0.001	—	%/°C
	Ref Supply coefficient	drift over [1.6, 3.6]V	—	-0.35/+0.35	—	%/V

Note: These values are based on characterization.

15.10.8 Temperature Sensor Characteristics

Table 15-27. Temperature Sensor Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ts_{Vout}	Temperature sensor output voltage	T= 25°C, V_{CC}=3.0V	-	656	-	mV
Ts_{Slope}	Temperature sensor slope	V_{CC} = 3.0V	-	2.24	-	mV/°C
$Ts_{V_{OUT}OV_{DDANA}}$	Variation over V_{DDANA} voltage	T= 25°C	-	1.1	-	mV/V

Note: 1.These values are based on characterization.

15.11 NVM Characteristics

Table 15-28. NVM Max Speed Characteristics

	Conditions	CPU Fmax (MHz)			
		0WS	1WS	2WS	3WS
PL0 (-40/85°C)	V_{DDIN}>1.6 V	6	12	12	12
	V_{DDIN}>2.7 V	7.5	12	12	12
PL2 (-40/85°C)	V_{DDIN}>1.6 V	14	28	42	48
	V_{DDIN}>2.7 V	24	45	48	48

Table 15-29. NVM Timing Characteristics

Symbol	Timings	Max	Units
t_{FPP}	Page Write(1)	2.5	ms
t_{FRE}	Row erase(1)	6	

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. For this Flash technology, a maximum number of 8 consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 15-30. NVM Reliability Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ret_{NVM25k}	Retention after up to 25k	Average ambient 55°C	10	50	-	Years
$Ret_{NVM2.5k}$	Retention after up to 2.5k	Average ambient 55°C	20	100	-	Years
Ret_{NVM100}	Retention after up to 100	Average ambient 55°C	25	>100	-	Years
Cyc_{NVM}	Cycling Endurance(1)	-40°C < Ta < 85°C	25K	100K	-	Cycles

Note: 1. An endurance cycle is a write and an erase operation.

Table 15-31. EEPROM Emulation(1) Reliability Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ret_{EE100k}	Retention after up to 100k	Average ambient 55°C	10	50	-	Years
Ret_{EE10k}	Retention after up to 10k	Average ambient 55°C	20	100	-	Years
Cyc_{EE}	Cycling Endurance(2)	-40°C < Ta < 85°C	100K	400K	-	Cycles

Note:

(1) The EEPROM emulation is a software emulation described in the App note AT03265.

(2) An endurance cycle is a write and an erase operation.

Table 15-32. Flash Erase and Programming Current

Symbol	Parameter	Typ.	Units
IDD_{NVM}	Maximum current (peak) during whole programming or erase operation	10	mA

15.12 Oscillators Characteristics

15.12.1 Crystal Oscillator (XOSC) Characteristics

Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN.

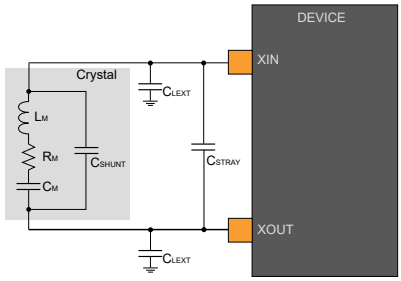
Table 15-33. Digital Clock Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units
F_{XIN}	XIN clock frequency	-	-	24	MHz
DC_{XIN}	XIN clock duty cycle	40	50	60	%

Crystal Oscillator Characteristics

The following Table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT.

Figure 15-3. Oscillator Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the Table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{STRAY} - SHUNT)$,

where C_{STRAY} is the capacitance of the pins and the PCB, and C_{SHUNT} is the shunt capacity of the crystal.

Table 15-34. Multi Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
FOUT	Crystal oscillator frequency		0.4	-	32	MHz
ESR	Crystal Equivalent Series Resistance - SF=3	F=0.4MHz, C_L=100pF XOSC, GAIN=0	-	-	5.6K	Ω
		F=2MHz, C_L=20pF XOSC,GAIN=0	-	-	416	
		F=4MHz, C_L=20pF XOSC, GAIN=1	-	-	243	
		F=8MHz, C_L=20pF XOSC, GAIN=2	-	-	138	
		F=16MHz, C_L=20pF XOSC, GAIN=3	-	-	66	
		F=32MHz, C_L=20pF XOSC, GAIN=4	-	-	56	

.....continued

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
C_{XIN}	Parasitic load capacitor		-	5.8	-	pF
C_{XOUT}			-	3.2	-	
T_{START}	Startup time	F=2MHz, C_L=20pF XOSC, GAIN=0	-	14k	48k	Cycles
		F=4MHz, C_L=20pF XOSC, GAIN=1	-	6.8k	19.5k	
		F=8MHz, C_L=20pF XOSC, GAIN=2	-	5.6k	13k	
		F=16MHz, C_L=20pF XOSC, GAIN=3	-	6.8k	14.5k	
		F=32MHz, C_L=20pF XOSC, GAIN=4	-	5.3K	9.6k	

Note: (1) These values are based on characterization.

Table 15-35. Power Consumption(1)

Symbol	Parameter	Conditions		Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F=2MHz - CL=20pF XOSC,GAIN=0, VCC=3.3V	AGC=OFF	Max 85°C Typ 25°C	-	89	133	µA
			AGC=ON		-	82	130	
		F=4MHz - CL=20pF XOSC,GAIN=1, VCC=3.3V	AGC=OFF		-	140	194	
			AGC=ON		-	102	156	
		F=8MHz - CL=20pF XOSC,GAIN=2, VCC=3.3V	AGC=OFF		-	243	313	
			AGC=ON		-	166	232	
		F=16MHz - CL=20pF XOSC,GAIN=3, VCC=3.3V	AGC=OFF		-	493	605	
			AGC=ON		-	293	393	
		F=32MHz - CL=20pF XOSC,GAIN=4, VCC=3.3V	AGC=OFF		-	1467	1777	
			AGC=ON		-	651	1045	

Note: (1) These values are based on characterization.

15.12.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

Digital Clock Characteristics
The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32 pin.

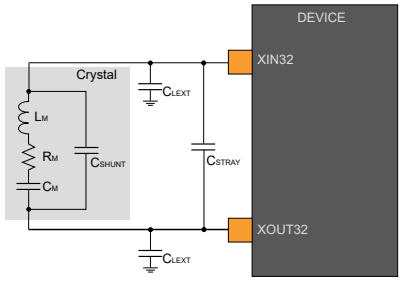
Table 15-36. Digital Clock Characteristics(1)

Symbol	Parameter	Min.	Typ.	Max.	Units
$f_{CPXIN32}$	XIN32 clock frequency		32.768	1000	kHz
DC_{XIN}	XIN32 clock duty cycle	40	50	60	%

Note: 1. These values are based on characterization.

Crystal Oscillator Characteristics
The following section describes the characteristics for the oscillator when a crystal is connected between XIN32 and XOUT32 pins.

Figure 15-4. Oscillator Crystal Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{STRAY} - SHUNT)$,

where C_{STRAY} is the capacitance of the pins and PCB, C_{SHUNT} is the shunt capacitance of the crystal.

Table 15-37. 32KHz Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Crystal oscillator frequency	-	-	32.768	-	kHz
CL	Crystal load capacitance	-	7	9	12.5	pF
C_{SHUNT}	Crystal shunt capacitance	-	0.6	-	2	pF
C_M	Motional capacitance	-	0.6	-	3	fF
ESR	Crystal Equivalent Series Resistance - SF=3	f=32.768kHz, C_L=12.5pF	-	50	70	kΩ
C_{XIN32k}	Parasitic load capacitor(2)	-	-	2.31	-	pF
$C_{XOUT32k}$		-	-	2.53	-	
$t_{STARTUP}$	Startup time	f=32.768kHz, C_L=12.5pF	-	25k	82k	Cycles
P_{on}	Drive Level (1)	-	-	-	0.1	µW

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. For device revision A, see Erratum 13425.

Table 15-38. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 85°C Typ 25°C	-	311	723	nA

Note: (1) These values are based on characterization.

15.12.3 32.768kHz Internal Oscillator (OSC32K) Characteristics

Table 15-39. 32KHz RC Oscillator Electrical Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	32.572	32.768	33.044	kHz
		at 25°C, over [1.8, 3.63]V	31.425	32.768	33.603	kHz
		over[-40,+85]°C, over [1.8, 3.63]V	28.581	32.768	34.716	kHz
I_{OSC32k}	Current consumption		-	0.54		µA
$T_{STARTUP}$	Startup time		-	1	2	cycles
Duty	Duty Cycle		-	50	-	%

Note: 1. These values are based on characterization.

Table 15-40. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 85°C Typ 25°C	-	0.54	1.10	µA

Note: (1) These values are based on characterization.

15.12.4 Internal Ultra Low Power 32KHz RC Oscillator (OSCULP32K) Characteristics

Table 15-41. Ultra Low Power Internal 32KHz RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	31.775	32.768	34.033	kHz
		at 25°C, over [1.8, 3.63]V	31.848	32.768	34.202	kHz
		over[-40,+85]°C, over [1.8, 3.63]V	26.296	32.768	38.384	kHz
Duty	Duty Cycle		-	50	-	%

15.12.5 Digital Frequency Locked Loop (DFLL48M) Characteristics

Table 15-42. DFLL48M Characteristics - Open Loop Mode(1,2)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{OpenOUT}$	Output frequency	DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE=512	46.6	47.8	49	MHz
$T_{OpenSTARTUP}$	Startup time	DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE=512 F_{OUT} within 90% of final value	-	8.3	9.1	µs

Notes:
1. DFLL48 in open loop can be used only with LDO regulator.
2. These values are based on characterization.

Table 15-43. DFLL48M Characteristics - Closed Loop Mode

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{CloseOUT}$	Average Output frequency	f_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464	47.963	47.972	47.981	MHz
F_{REF}(2,3)	Input reference frequency		732	32768	33000	Hz
$F_{CloseJitter}$(1)	Period Jitter	f_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464	-	-	0.51	ns
T_{Lock}(1)	Lock time	F_{REF} = XTAL, 32.768kHz, 100ppm DFLLMUL=1464 DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE = 512 DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.CCDIS = 1 DFLLMUL.FSTEP = 10		200	700	µs

Notes:

1. These values are based on characterization.
2. To insure that the device stays within the maximum allowed clock frequency, any reference clock for the DFLL in close loop must be within a 2% error accuracy.
3. These values are based on simulation. They are not covered by production test limits or characterization.

Table 15-44. DFLL48M Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Power consumption Open Loop	DFLLVAL.COARSE=DFLL48M_COARSE_CAL	Max.85°C Typ.25°C	-	286	-	µA
I_{DD}	Power consumption Closed Loop	F_{REF} = 32.768kHz, V_{CC}=3.3V		-	362	-	µA

Note:

1. These values are based on characterization.

15.12.6 16MHz RC Oscillator (OSC16M) Characteristics

Table 15-45. Multi RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	V_{CC}=3.3V, T=25°C	3.953	4	4.062	MHz
			7.877	8	8.112	
			11.857	12	12.139	
			15.754	16	16.235	
TempDrift	Freq vs. temperature drift		-4		4	%
SupplyDrift	Freq vs. supply drift		-2		2	
T_{WUP}(2)	Wake up time - 1st clock edge after enable	F_{OUT} = 4MHz	-	0.12	0.27	µs
		F_{OUT} = 8MHz	-	0.12	0.25	
		F_{OUT} = 12MHz	-	0.12	0.27	
		F_{OUT} = 16MHz	-	0.12	0.25	

.....continued						
Symbol	**Parameter**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Units**
$T_{STARTUP}$(2)	Startup time	F_{OUT} = 4MHz	-	1.2	2.9	µs
		F_{OUT} = 8MHz	-	1.3	2.6	
		F_{OUT} = 12MHz	-	1.3	2.8	
		F_{OUT} = 16MHz	-	1.4	3.1	
Duty(1)	Duty Cycle	-	45	50	55	%

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 15-46. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	F_{out}=4MHz, V_{CC}=3.3V	Max.85°C Typ.25°C	-	64	96	µA
		F_{out}=8MHz, V_{CC}=3.3V		-	91	122	
		F_{out}=12MHz, V_{CC}=3.3V		-	114	144	
		F_{out}=16MHz, V_{CC}=3.3V		-	141	169	

Note:

1. These values are based on characterization.

15.12.7 Digital Phase Lock Loop (DPLL) Characteristics

Table 15-47. Fractional Digital Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{IN}	Input Clock Frequency		32	-	2000	kHz
F_{OUT}	Output frequency	PL0	48	-	48	MHz
		PL2	48	-	96	MHz
J_P(2)	Period Jitter	PL0, F_{IN}=32kHz @ F_{OUT}=48MHz	-	1.9	5.0	%
		PL2, F_{IN}=32kHz @ F_{OUT}=48MHz	-	1.9	4.0	
		PL2, F_{IN}=32kHz @ F_{OUT}=96MHz	-	3.3	7.0	
		PL0, F_{IN}=2MHz @ F_{OUT}=48MHz	-	2.0	8.0	
		PL2, F_{IN}=2MHz @ F_{OUT}=48MHz	-	2.0	4.0	
		PL2, F_{IN}=2MHz @ F_{OUT}=96MHz	-	4.2	7.0	
T_{LOCK}(2)	Lock Time	After startup, time to get lock signal, F_{IN}=32kHz @ F_{OUT}=96MHz	-	1	2	ms
		After startup, time to get lock signal, F_{IN}=2MHz @ F_{OUT}=96MHz	-	25	35	µs
Duty(1)	Duty Cycle		40	50	60	%

Notes:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 15-48. Power Consumption(1)

Symbol	Parameter	Conditions	TA	Min.	Typ.	Max.	Units
I_{DD}	Current Consumption	Ck=48MHz (PL0)	Max.85°C Typ.25°C	-	454	548	µA
		Ck=96MHz (PL2)		-	934	1052	

Note:

1. These values are based on characterization.

15.13 Timing Characteristics

15.13.1 External Reset Pin

Table 15-49. External Reset Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
t_{EXT}	Minimum reset pulse width		1	-	-	µs

15.13.2 SERCOM in SPI Mode in PL0

Table 15-50. SPI Timing Characteristics and Requirements(1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
t_{SCK}	SCK period	Master, VDD>2.70V		141	153		ns
		Master, VDD>1.8V		147	159		
t_{SCKW}	SCK high/low width	Master		-	0.5*tSCK	-	
t_{SCKR}	SCK rise time(2)	Master		-	0.25*tSCK	-	
t_{SCKF}	SCK fall time(2)	Master		-	0.25*tSCK	-	
t_{MIS}	MISO setup to SCK	Master, VDD>2.70V		141			
		Master, VDD>1.8V		147			
t_{MIH}	MISO hold after SCK	Master, VDD>2.70V		0			
		Master, VDD>1.8V		0			
t_{MOS}	MOSI setup SCK	Master, VDD>2.70V				30	
		Master, VDD>1.8V				30.6	
t_{MOH}	MOSI hold after SCK	Master, VDD>2.70V		-9			
		Master, VDD>1.8V		-8.5			
t_{SSCK}	Slave SCK Period	Slave, VDD>2.70V		220	250		
		Slave, VDD>1.8V		230	250	-	
t_{SSCKW}	SCK high/low width	Slave		-	0.5*tSCK	-	
t_{SSCKR}	SCK rise time(2)	Slave		-	0.25*tSCK	-	
t_{SSCKF}	SCK fall time(2)	Slave		-	0.25*tSCK	-	
t_{SIS}	MOSI setup to SCK	Slave, VDD>2.70V		42	-	-	
		Slave, VDD>1.8V		42			
t_{SIH}	MOSI hold after SCK	Slave, VDD>2.70V		0			
		Slave, VDD>1.8V		0			
t_{SSS}	SS setup to SCK	Slave	PRELOADEN=1				
			PRELOADEN=0				
t_{SSH}	SS hold after SCK	Slave					
t_{SOS}	MISO setup before SCK	Slave, VDD>2.70V				109	
		Slave, VDD>1.8V				115	
t_{SOH}	MISO hold after SCK	Slave, VDD>2.70V		17.3			
		Slave, VDD>1.8V		17.3			
t_{SOSS}	MISO setup after SS low	Slave, VDD>2.70V				95	
		Slave, VDD>1.8V				102	
t_{SOSH}	MISO hold after SS high	Slave, VDD>2.70V		10.2			ns
		Slave, VDD>1.8V		10.2			

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Figure 15-5. SPI Timing Requirements in Master Mode

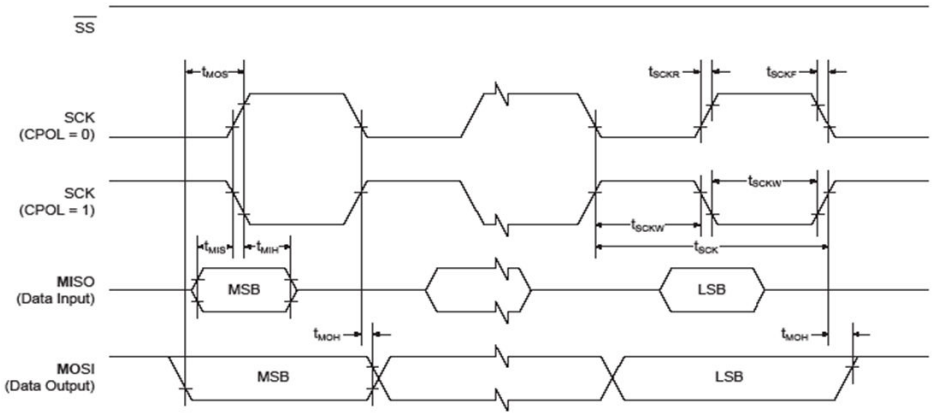
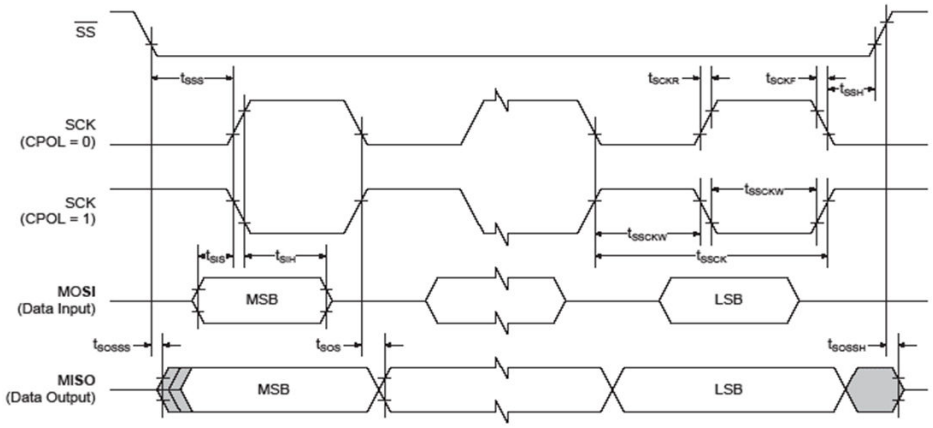


Figure 15-6. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

- Master Mode
  $f_{SCKmax} = 1/2*(t_{MIS} + t_{valid})$, where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12ns Max, $f_{SPCKMax}$ = 3.7MHz @ VDDIO > 2.7V
- Slave Mode
  $f_{SCKmax} = 1/2*(t_{SOV} + t_{su})$, where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 6MHz @ VDDIO > 2.7V

15.13.3 SERCOM in SPI Mode in PL2

Table 15-51. SPI Timing Characteristics and Requirements(1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
t_{SCK}	SCK period	Master, VDD>2.70V		41.1	53		ns
		Master, VDD>1.8V		52.6	65		
t_{SCKW}	SCK high/low width	Master		-	0.5*t_{SCK}	-	
t_{SCKR}	SCK rise time(2)	Master		-	0.25*t_{SCK}	-	
t_{SCKF}	SCK fall time(2)	Master		-	0.25*t_{SCK}	-	
t_{MIS}	MISO setup to SCK	Master, VDD>2.70V		41.1			
		Master, VDD>1.8V		52.6			
t_{MIH}	MISO hold after SCK	Master, VDD>2.70V		0			
		Master, VDD>1.8V		0			
t_{MOS}	MOSI setup SCK	Master, VDD>2.70V				8.5	
		Master, VDD>1.8V				13.1	
t_{MOH}	MOSI hold after SCK	Master, VDD>2.70V		0.5			
		Master, VDD>1.8V		1			
t_{SSCK}	Slave SCK Period	Slave, VDD>2.70V		74	90	-	
		Slave, VDD>1.8V		95	110	-	
t_{SSCKW}	SCK high/low width	Slave		-	0.5*t_{SCK}	-	
t_{SSCKR}	SCK rise time(2)	Slave		-	0.25*t_{SCK}	-	
t_{SSCKF}	SCK fall time(2)	Slave		-	0.25*t_{SCK}	-	
t_{SIS}	MOSI setup to SCK	Slave, VDD>2.70V		10.3	-	-	
		Slave, VDD>1.8V		11.8	-	-	
t_{SIH}	MOSI hold after SCK	Slave, VDD>2.70V		0	-	-	
		Slave, VDD>1.8V		0	-	-	
t_{SSS}	SS setup to SCK	Slave	PRELOADEN=1				
			PRELOADEN=0				
t_{SSH}	SS hold after SCK	Slave					
t_{SOS}	MISO setup before SCK	Slave, VDD>2.70V		-	-	36.9	ns
		Slave, VDD>1.8V		-	-	47.5	
t_{SOH}	MISO hold after SCK	Slave, VDD>2.70V		11.5	-	-	
		Slave, VDD>1.8V		11.5	-	-	
t_{SOSS}	MISO setup after SS low	Slave, VDD>2.70V		-	-	31	
		Slave, VDD>1.8V		-	-	41.3	
t_{SOSH}	MISO hold after SS high	Slave, VDD>2.70V		6.2	-	-	
		Slave, VDD>1.8V		6.2	-	-	

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Figure 15-7. SPI Timing Requirements in Master Mode

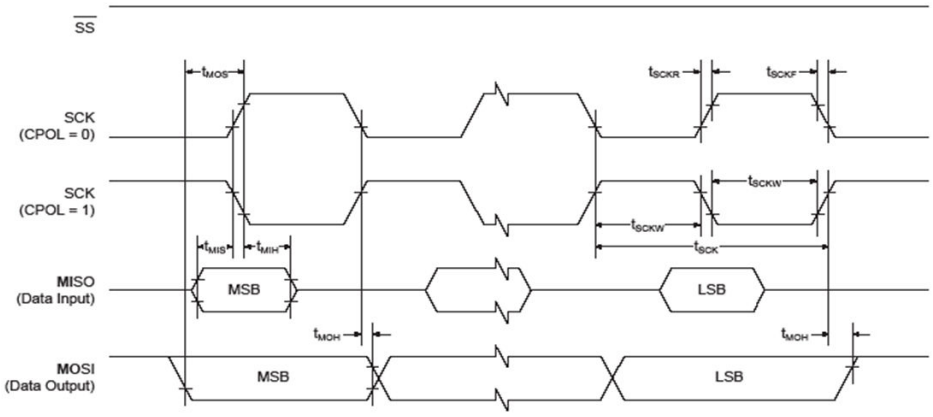
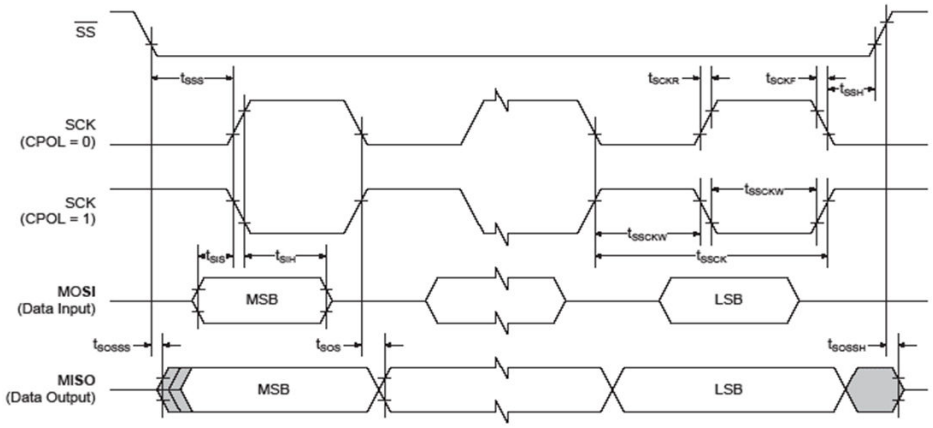


Figure 15-8. SPI Timing Requirements in Slave Mode



Maximum SPI Frequency

- Master Mode
  f_{SCKmax} = 1/2*(t_{MIS} + t_{valid}), where t_{valid} is the slave time response to output data after detecting an SCK edge. For a non-volatile memory with t_{valid} = 12ns Max, $f_{SPCKMax}$ = 9.8MHz @ VDDIO > 2.7V
- Slave Mode
  f_{SCKmax} = 1/2*(t_{SOV} + t_{su}), where t_{su} is the setup time from the master before sampling data. With a perfect master (t_{su}=0), $f_{SPCKMax}$ = 16.3MHz @ VDDIO > 2.7V

15.14 USB Characteristics

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

The USB interface is USB-IF certified:

- TID 40001709 - Peripheral Silicon > Low/Full Speed > Silicon Building Blocks
- TID 120000459 - Embedded Hosts > Full Speed

Electrical configuration required to be USB-compliant:

- the performance level must be PL2 only
- the CPU frequency must be higher 8MHz when USB is active (No constraint for USB suspend mode)
- the operating voltages must be 3.3V (Min. 3.0V, Max. 3.6V).
- the GCLK_USB frequency accuracy source must be less than:
  – in USB device mode, 48MHz +/-0.25%

– in USB host mode, 48MHz +/-0.05%

Table 15-52. GCLK_USB Clock Setup Recommendations

Clock setup		USB Device	USB Host
DFLL48M	Open loop	No	No
	Close loop, Ref. internal OSC source	No	No
	Close loop, Ref. external XOSC source	Yes	No
	Close loop, Ref. SOF (USB recovery mode)(1)	Yes(2)	N/A
FDPLL	internal OSC (32K, 8M…)	No	No
	external OSC (<1MHz)	Yes	No
	external OSC (>1MHz)	Yes(3)	Yes

Notes:

1. When using DFLL48M in USB recovery mode, the Fine Step value must be 0xA to guarantee a USB clock at +/-0.25% before 11ms after a resume. Only usable in LDO regulator mode.
2. Very high signal quality and crystal less. It is the best setup for USB Device mode.
3. FDPLL lock time is short when the clock frequency source is high (> 1 MHz). Thus, FDPLL and external OSC can be stopped during USB suspend mode to reduce consumption and guarantee a USB wake-up time (See TDRSMDN in USB specification).

15.15 AT86RF212B Characteristics

15.15.1 Digital Interface Timing Characteristics

Note: This is for internal communication between the AT86RF212B transceiver and the SAM L21 microcontroller.

Test Conditions: T_{OP} = +25°C, V_{DD} = 3.0V, C_{Load} = 50pF (unless otherwise stated).

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
f_{sync}	SCLK frequency	Synchronous operation			8	MHz
f_{async}	SCLK frequency	Asynchronous operation			7.5	MHz
t_1	/SEL falling edge to MISO active				180	ns
t_2	SCLK falling edge to MISO out	Data hold time	25			ns
t_3	MOSI setup time		10			ns
t_4	MOSI hold time		10			ns
t_5	LSB last byte to MSB next byte		250(1)			ns
t_6	/SEL rising edge to MISO tri state				10	ns
t_7	SLP_TR pulse width	TX start trigger	62.5		Note(2)	ns
t_8	SPI idle time: SEL rising to falling edge	SPI read/write, standard SRAM and frame access modes Idle time between consecutive SPI accesses	250			ns

.....continued						
Symbol	**Parameter**	**Condition**	**Min.**	**Typ.**	**Max.**	**Unit**
t_{8a}	SPI idle time: SEL rising to falling edge	Fast SRAM read/write access mode Idle time between consecutive SPI accesses	500			ns
t_9	SCLK rising edge LSB to /SEL rising edge			250		ns
t_{10}	Reset pulse width	≥ 10 clock cycles at 16MHz	625			ns
t_{11}	SPI access latency after reset	≥ 10 clock cycles at 16MHz	625			ns
t_{12}	Dynamic frame buffer protection: IRQ latency			750		ns
t_{IRQ}	IRQ_2, IRQ_3, IRQ_4 latency	Relative to the event to be indicated		9		µs
f_{CLKM}	Output clock frequency at pin 17 (CLKM)	Configurable by the CLKM_CTRL bits in the TRX_CTRL 0 register (TRX_CTRL_0.CLKM_CTRL) CLKM_CTRL = 0		0		MHz
		CLKM_CTRL = 1		1		MHz
		CLKM_CTRL = 2		2		MHz
		CLKM_CTRL = 3		4		MHz
		CLKM_CTRL = 4		8		MHz
		CLKM_CTRL = 5		16		MHz
		CLKM_CTRL = 6		250		kHz
		CLKM_CTRL = 7[3]		20.0		kHz
		CLKM_CTRL = 7[4]		40.0		kHz
		CLKM_CTRL = 7[5]		25.0		kHz
		CLKM_CTRL = 7[6]		62.5		kHz

1. For Fast SRAM read/write accesses on address space 0x82 – 0x94 the time t_5(Min.) and t_8(Min.) increases to 500ns.
2. Maximum pulse width less than (TX frame length + 16µs).
3. 1/50MHz; Only in BPSK mode with fPSDU = 20kb/s.
4. 1/25MHz; Only in BPSK mode with fPSDU = 40kb/s.
5. 1/40MHz; Only in O-QPSK mode with fPSDU = 100/200/400kb/s.
6. 1/16MHz; Only in O-QPSK mode with fPSDU = 250/500/1000kb/s.

Related Links

14.8.3 TRX_CTRL_0

15.15.2 General RF Specifications

Test Conditions (unless otherwise stated):

V_{DD} = 3.0V, f_{RF} = 914MHz, T_{OP} = +25°C, Measurement setup in *Basic Application Schematic*.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
f_{RF}	Frequency range	As specified in [1]	868.3	914	924	MHz
		1MHz spacing	769		935	MHz
		100kHz spacing	769.0		794.5	MHz
		100kHz spacing	857.0		882.5	MHz
		100kHz spacing	902.0		928.5	MHz
f_{CH}	Channel spacing	As specified in [1] except CHANNEL = 0		2		MHz
		1MHz spacing		1000		kHz
		100kHz spacing		100		kHz
f_{CHIP}	Chip rate	BPSK as specified in [1](1)		300		kchip/s
		BPSK as specified in [1](2)		600		kchip/s
		O-QPSK as specified in [2](1)		400		kchip/s
		O-QPSK as specified in [2], [3](2)		1000		kchip/s
f_{HDR}	Header bit rate (SHR, PHR)	BPSK as specified in [1](1)		20		kb/s
		BPSK as specified in [1](2)		40		kb/s
		O-QPSK as specified in [2](1)		100		kb/s
		O-QPSK as specified in [2], [3](2)		250		kb/s
f_{PSDU}	PSDU bit rate	BPSK as specified in [1](1)		20		kb/s
		BPSK as specified in [1](2)		40		kb/s
		O-QPSK as specified in [2](1)		100		kb/s
		O-QPSK as specified in [2], [3](2)		250		kb/s
		OQPSK_DATA_RATE = 1(1)		200		kb/s
		OQPSK_DATA_RATE = 2(1)		400		kb/s
		OQPSK_DATA_RATE = 1(2)		500		kb/s
		OQPSK_DATA_RATE = 2(2)		1000		kb/s
f_{CLK}	Crystal oscillator frequency	Reference oscillator		16		MHz
f_{SRD}	Symbol rate deviation	PSDU bit rate				
	Reference frequency accuracy for correct functionality	20/40/100/250kb/s	-60(3)		+60	ppm
		200/400/500/1000kb/s	-40		+40	ppm

1. For TRX_CTRL_2.SUB_MODE = 0.
2. For TRX_CTRL_2.SUB_MODE = 1.
3. A reference frequency accuracy of ±40ppm is required by [1], [2], [3], [4].

Related Links

23. References
14.8.12 TRX_CTRL_2

15.15.3 Transmitter Characteristics

Test Conditions (unless otherwise stated):

V_{DD} = 3.0V, f_{RF} = 914MHz, T_{OP} = +25°C, Measurement setup in *Basic Application Schematic*.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
P_{TX_MAX}	TX Output power	Maximum configurable TX output power value				
		Normal mode		+5		dBm
		Boost mode		+10	+11	dBm
P_{RANGE}	Output power range	36 steps, configurable in the PHY_TX_PWR register		35		dB
P_{ACC}	Output power tolerance	868.3MHz			±3	dB
P_{1dB}	1dB compression point	Normal mode		5		dBm
		Boost mode		10		dBm
EVM	Error vector magnitude(1)	BPSK-20		5		%rms
		BPSK-40		8		%rms
		BPSK-40-ALT		8		%rms
		OQPSK-SIN-RC-100(2)(3)		29		%rms
		OQPSK-SIN-250		16		%rms
		OQPSK-RC-100(3)		10		%rms
		OQPSK-RC-250		10		%rms
P_{2nd_HARM}	2nd Harmonics(4)	TX power: +10dBm				
		914MHz		–27		dBm
		868.3MHz		–26		dBm
		782MHz		–28		dBm
		TX power: -2dBm				
		914MHz		–53		dBm
		868.3MHz		–46		dBm
		782MHz		–42		dBm
P_{3rd_HARM}	3rd Harmonics(4)	TX power: +10dBm				
		914MHz		–22		dBm
		868.3MHz		–22		dBm
		782MHz		–23		dBm
		TX power: -2dBm				
		914MHz		–38		dBm
		868.3MHz		–38		dBm
		782MHz		–37		dBm
P_{SPUR_TX}	Spurious Emissions(5)	30 – ≤ 1000MHz			-36	dBm
		>1 – 12.75GHz			-30	dBm

1. Power settings according to the TX_PWR bits in the PHY_TX_PWR register (PHY_TX_PWR.TX_PWR).
2. The EVM of OQPSK-SIN-RC-100 is significantly higher than the EVM of the other modulation schemes. This phenomenon can be explained by the fact that the combination of SIN and RC shaping as specified in IEEE 802.15.4-2006/2011 inherently show some inter-chip interference.
3. The EVM is valid up to +5dBm.
4. Measured single ended @ RFP/ RFN into 50Ω; termination of the other pin with 50Ω; constant wave signal.
5. Complies with EN 300 220, FCC-CFR-47 part 15, ARIB STD-108, RSS-210.

Related Links

14.8.5 PHY_TX_PWR

15.15.4 Receiver Characteristics

Test Conditions (unless otherwise stated):

V_{DD} = 3.0V, f_{RF} = 914MHz, T_{OP} = +25°C, Measurement setup in *Basic Application Schematic*.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
P_{SENS}	Receiver sensitivity	f_{RF}= 868.3MHz				
		BPSK-20(1)(3)		-110		dBm
		OQPSK-SIN-RC-100(1)(4)		-101		dBm
		OQPSK-SIN-RC-200(2)		-99		dBm
		OQPSK-SIN-RC-400(2)		-91		dBm
		OQPSK-RC-100(1)		-102		dBm
		OQPSK-RC-200(2)		-100		dBm
		OQPSK-RC-400(2)		-97		dBm
		f_{RF}= 914MHz				
		BPSK-40(1)(3)		-108		dBm
		OQPSK-SIN-250(1)(4)		-100		dBm
		OQPSK-SIN-500(2)		-98		dBm
		OQPSK-SIN-1000(2)		-93		dBm
		f_{RF}= 782MHz				
		OQPSK-RC-250(1)(5)		-101		dBm
		OQPSK-RC-500(2)		-99		dBm
		OQPSK-RC-1000(2)		-95		dBm
RL_{RX}	RX Return loss	100Ω differential impedance		12		dB
NF	Noise figure			7		dB
P_{RX_MAX}	Maximum RX input level(1)			7	10	dBm
P_{CRSB20}	Channel rejection/selectivity: BPSK-20(3)	f_{RF}= 868.3MHz				
		P_{RF}= -89dBm(1)				
		-2MHz		39		dB
		-1MHz		33		dB
		+1MHz		19		dB
		+2MHz		39		dB

.....continued

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
$P_{CRSO100}$	Channel rejection/selectivity: OQPSK-SIN-RC-100(4)	f_{RF}= 868.3MHz				
		P_{RF}= -82dBm(1)				
		-2MHz		35		dB
		-1MHz		24		dB
		+1MHz		17		dB
		+2MHz		35		dB
P_{ACRB40}	Adjacent channel rejection: BPSK-40(3)	P_{RF}= -89dBm(1)				
		-2MHz		38		dB
		+2MHz		38		dB
$P_{AACRB40}$	Alternate channel rejection: BPSK-40(3)	P_{RF}= -89dBm(1)				
		-4MHz		56		dB
		+4MHz		56		dB
$P_{ACROS250}$	Adjacent channel rejection: OQPSK-SIN-250(4)	PRF= -82dBm(1)				
		-2MHz		30(6)		dB
		+2MHz		30(6)		dB
$P_{AACROS250}$	Alternate channel rejection: OQPSK-SIN-250(4)	P_{RF}= -82dBm(1)				
		-4MHz		47(6)		dB
		+4MHz		47(6)		dB
$P_{ACROR250}$	Adjacent channel rejection: OQPSK-RC-250(5)	P_{RF}= -82dBm(1)				
		-2MHz		32		dB
		+2MHz		32		dB
$P_{AACROR250}$	Alternate channel rejection: OQPSK-RC-250(5)	P_{RF}= -82dBm(1)				
		-4MHz		50		dB
		+4MHz		50		dB
RX_{BL}	Blocking	fRF= 868.3MHz				
		Refer to ETSI EN 300 220-1 PRF= -90dBm(1)				
		BPSK-20, ±2MHz		38		dB
		BPSK-20, ±10MHz		71		dB
		OQPSK-SIN-RC-100, ±2MHz		34		dB
		OQPSK-SIN-RC-100, ±10MHz		68		dB
P_{SPUR_RX}	Spurious emissions	LO leakage		-71		dBm
		30 – ≤ 1000MHz			-57	dBm
		>1 – 12.75GHz			-47	dBm

.....continued						
Symbol	**Parameter**	**Condition**	**Min.**	**Typ.**	**Max.**	**Unit**
IIP3	3rd– order intercept point	868.3MHz, at maximum gain Offset freq. interf. 1 = 2MHz Offset freq. interf. 2 = 4MHz		-12		dBm
IIP2	2nd– order intercept point	868.3MHz, at maximum gain Offset freq. interf. 1 = 3MHz Offset freq. interf. 2 = 4MHz		25		dBm
$RSSI_{TOL}$	RSSI tolerance	Tolerance within gain step			±6	dB
$RSSI_{RANGE}$	RSSI dynamic range			87		dB
$RSSI_{RES}$	RSSI resolution			3.1		dB
$RSSI_{BASE\_VAL}$	RSSI sensitivity	Defined as RSSI_BASE_VAL				
		BPSK with 300kchips/s		-100		dBm
		BPSK with 600kchips/s		-99		dBm
		O-QPSK with 400kchips/s, SIN and RC-0.2 shaping		-98		dBm
		O-QPSK with 400kchips/s, RC-0.2 shaping		-98		dBm
		O-QPSK with 1000kchips/s, SIN shaping		-98		dBm
		O-QPSK with 1000kchips/s, RC-0.8 shaping		-97		dBm
$RSSI_{MIN}$	Minimum RSSI value	PRF≤ RSSI_BASE_VAL		0		
$RSSI_{MAX}$	Maximum RSSI value	PRF≥ RSSI_BASE_VAL + 87dB		28		

1. AWGN channel, PER ≤ 1%, PSDU length 20 octets.
2. AWGN channel, PER ≤ 1%, PSDU length 127 octets.
3. Compliant to [1].
4. Compliant to [2].
5. Compliant to [4].
6. Channel rejection is limited by modulation side lobes of interfering signal.

Related Links

23. References

15.15.5 Current Consumption Specifications

Note: The current consumption numbers are only applicable for the RF transceiver, AT86RF212B.

Test Conditions (unless otherwise stated):

V_{DD} = 3.0V, f_{RF} = 914MHz, T_{OP} = +25°C, Measurement setup in *Basic Application Schematic*.

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
I_{BUSY_TX}	Supply current transmit state	North American band, O-QPSK modulation				
		PTX= +10dBm (boost mode)		26.5		mA
		PTX= +5dBm (normal mode)		18.0		mA
		PTX= +0dBm (normal mode)		13.5		mA
		PTX= -25dBm (normal mode)		9.5		mA
I_{RX_ON}	Supply current RX_ON state	North American band, O-QPSK modulation				
		high sensitivity RX_PDT_LEVEL = [0x0]		9.2		mA
		receiver desensitize RX_PDT_LEVEL = [0x1, ..., 0xE, 0xF](1)		8.7		mA
I_{PLL_ON}	Supply current PLL_ON state			5.0		mA
I_{TRX_OFF}	Supply current TRX_OFF state			450		µA

1. Refer to *Receiver Configuration*.
2. All power consumption measurements are performed with CLKM disabled.

Related Links

14.4.2.3 Configuration

15.15.6 Crystal Parameter Requirements

Test Conditions: T_{OP} = +25°C, V_{DD} = 3.0V (unless otherwise stated).

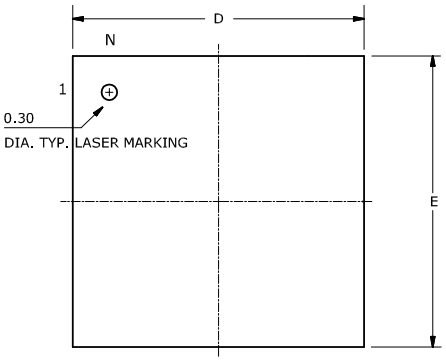
Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
f_0	Crystal frequency			16		MHz
C_L	Load capacitance		8		14	pF
C_0	Crystal shunt capacitance				7	pF
ESR	Equivalent series resistance				100	Ω

16. Packaging Information

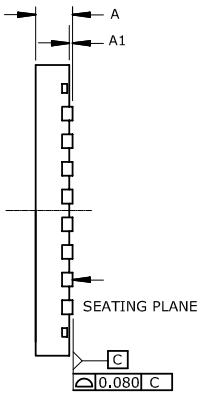
16.1 Package Drawings

16.1.1 32 pin QFN

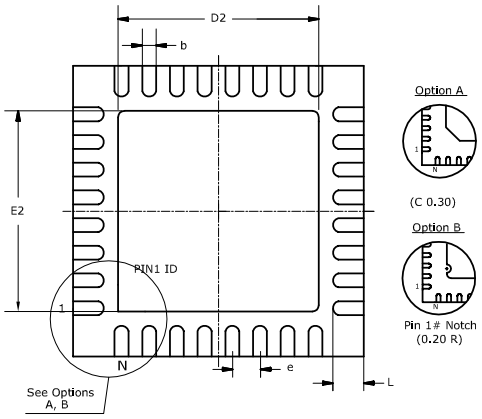
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	----	1.00	
A1	0.00	----	0.05	
D/E		5.00 BSC		
D2/E2	3.50	3.60	3.70	
L	0.30	0.40	0.50	
b	0.18	0.25	0.30	2
e		0.50 BSC		
n		32		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VHHD-5, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

10/05/2015

Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	TITLE PG, 32 Leads , 0.50mm pitch, 5x5x1.0mm Body Size Very Thin Quad Flat No Lead Package (VQFN) Sawn	GPC ZKV	DRAWING NO. PG	REV. C

Note: The exposed die attach pad is connected inside the device to GND and GNDANA.

Table 16-1. Device and Package Maximum Weight

90	mg

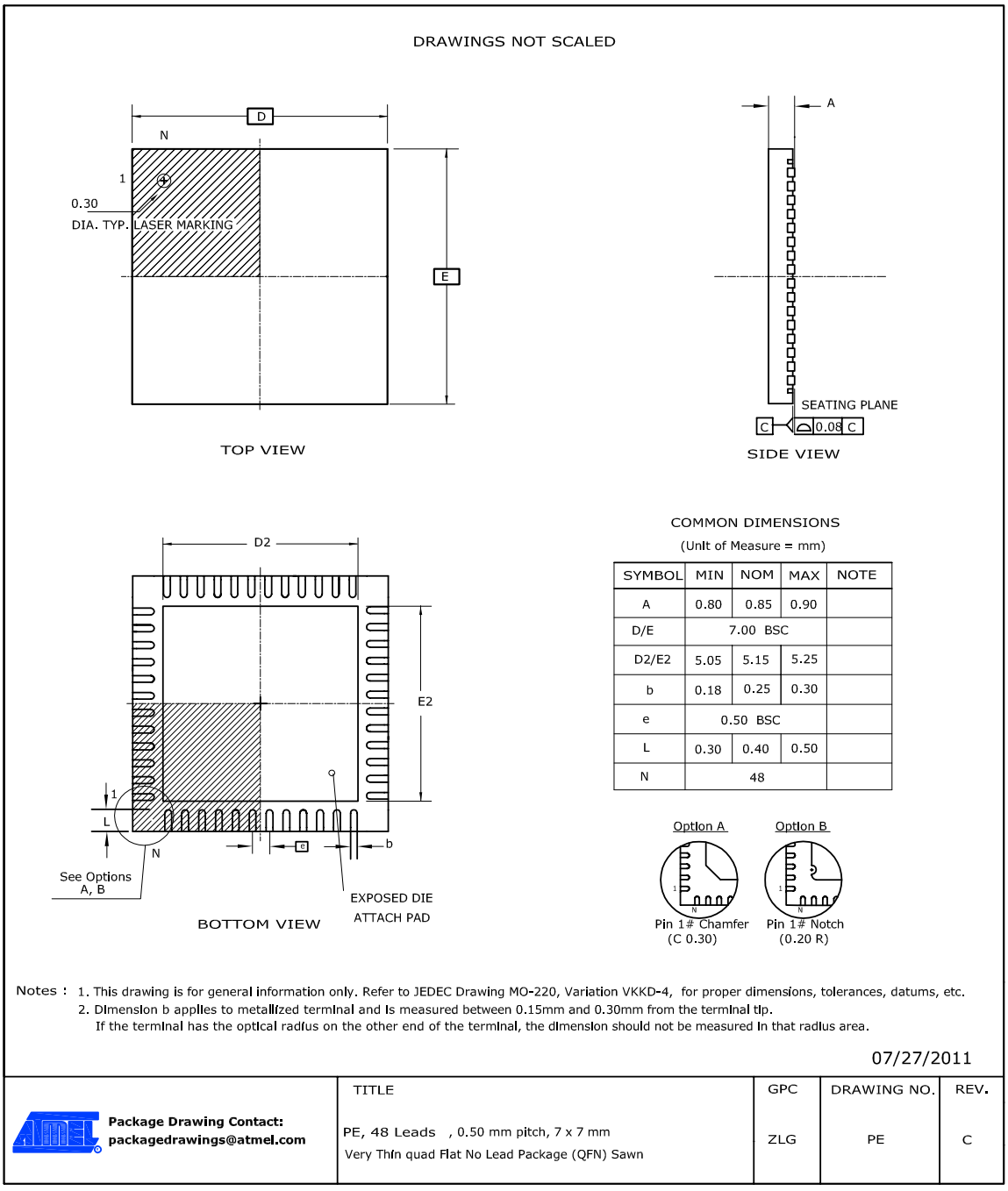
Table 16-2. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 16-3. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

16.1.2 48 pin QFN



SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.85	0.90	
D/E		7.00 BSC		
D2/E2	5.05	5.15	5.25	
b	0.18	0.25	0.30	
e		0.50 BSC		
L	0.30	0.40	0.50	
N		48		

Note: The exposed die attach pad is not connected electrically inside the device.

Table 16-4. Device and Package Maximum Weight

140	mg

Table 16-5. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 16-6. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

17. Schematic Checklist

17.1 Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a SAM R30 design. This chapter illustrates the recommended power supply connections, how to connect external analog references, programmer, debugger, oscillator and crystal.

17.1.1 Operation in Noisy Environment

If the device is operating in an environment with much electromagnetic noise it must be protected from this noise to ensure reliable operation. In addition to following best practice EMC design guidelines, the recommendations listed in the schematic checklist sections must be followed. In particular placing decoupling capacitors very close to the power pins, a RC-filter on the $\overline{\text{RESET}}$ pin, and a pull-up resistor on the SWCLK pin is critical for reliable operations. It is also relevant to eliminate or attenuate noise in order to avoid that it reaches supply pins, I/O pins and crystals.

17.2 Power Supply

The SAM R30 supports a single or dual power supply from 1.8V to 3.63V. The same voltage must be applied to both VDDIN and VDDANA.

The internal voltage regulator has four different modes:

- Linear mode: this mode does not require any external inductor. This is the default mode when CPU and peripherals are running
- Low Power (LP) mode: This is the default mode used when the chip is in standby mode
- Shutdown mode: When the chip is in backup mode, the internal regulator is turned off

Selecting between switching mode and linear mode can be done by software on the fly, but the power supply must be designed according to which mode is to be used.

17.2.1 Power Supply Connections

The following figures shows the recommended power supply connections for switched/linear mode, linear mode only and with battery backup.

Figure 17-1. Power Supply Connection for Linear Mode Only

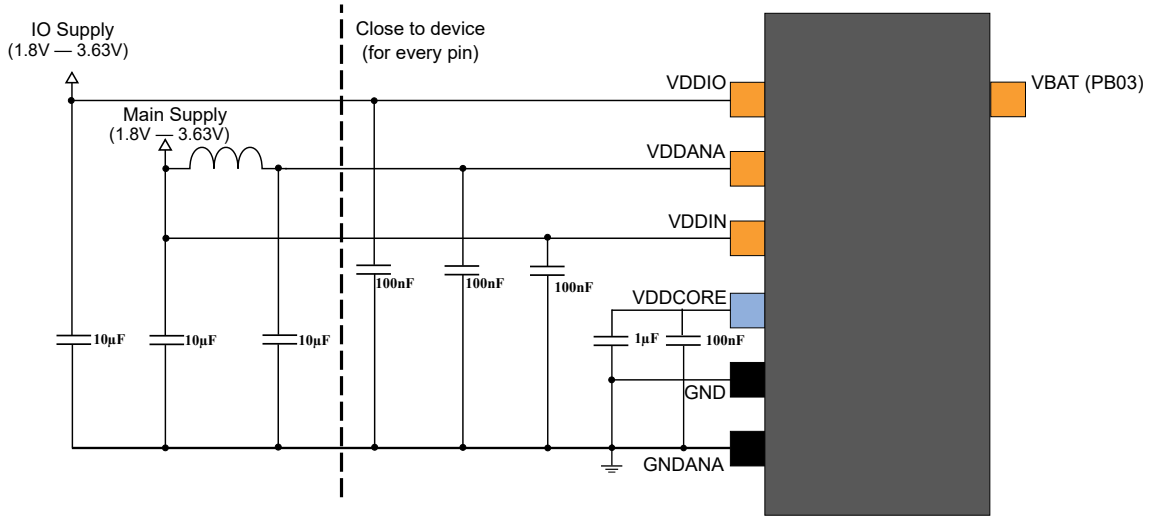


Figure 17-2. Power Supply Connection for Battery Backup

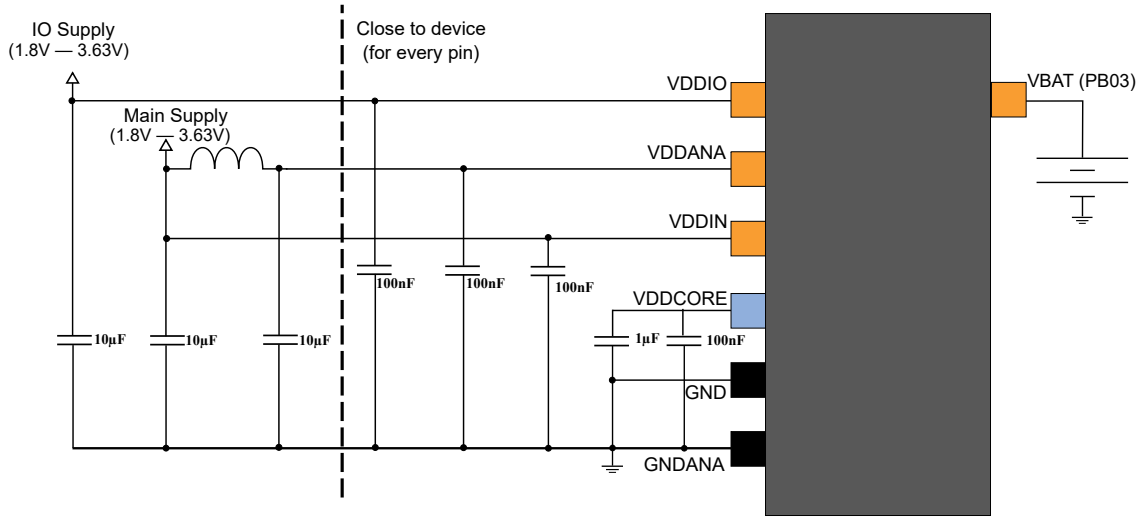


Table 17-1. Power Supply Connections, V_{DDCORE} From Internal Regulator

Signal Name	Recommended Pin Connection	Description
V_{DDIO}	1.8V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Decoupling/filtering inductor 10µH(1)(3)	Digital supply voltage
V_{DDANA}	1.8V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Ferrite bead(4) prevents the V_{DD} noise interfering with V_{DDANA}	Analog supply voltage
V_{DDIN}	1.8V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Decoupling/filtering inductor 10µH(1)(3)	Digital supply voltage
V_{BAT}	1.8V to 3.6V when connected	External battery supply input
V_{DDCORE}	0.9V to 1.2V typical Decoupling/filtering capacitors 100nF(1)(2) and 1µF(1)	Linear regulator mode: Core supply voltage output/ external decoupling pin
GND		Ground
GND_{ANA}		Ground for the analog power domain

1. These values are only given as a typical example.
2. Decoupling capacitors should be placed close to the device for each supply pin pair in the signal group, low ESR capacitors should be used for better decoupling.
3. An inductor should be added between the external power and the V_{DD} for power filtering.
4. A ferrite bead has better filtering performance compared to standard inductor at high frequencies. A ferrite bead can be added between the main power supply (V_{DD}) and V_{DDANA} to prevent digital noise from entering the analog power domain. The bead should provide enough impedance (e.g. 50Ω at 20MHz and 220Ω at 100MHz) to separate the digital and analog power domains. Make sure to select a ferrite bead designed for filtering applications with a low DC resistance to avoid a large voltage drop across the ferrite bead.

17.2.2 Special Considerations for QFN Packages

The QFN package has an exposed paddle that must be connected to GND.

17.3 External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the following circuits in Figure 17-3 and Figure 17-4 are not necessary.

Figure 17-3. External Analog Reference Schematic With Two References

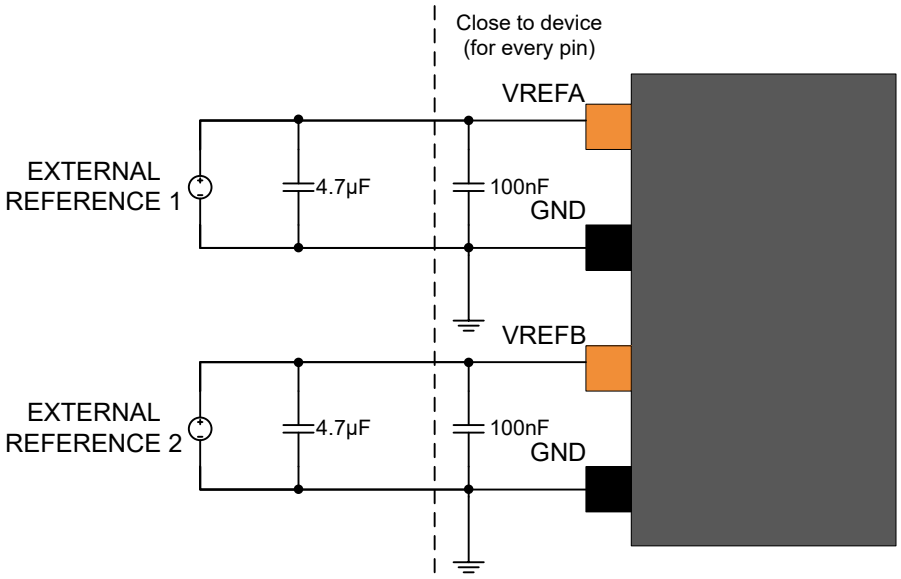


Figure 17-4. External Analog Reference Schematic With One Reference

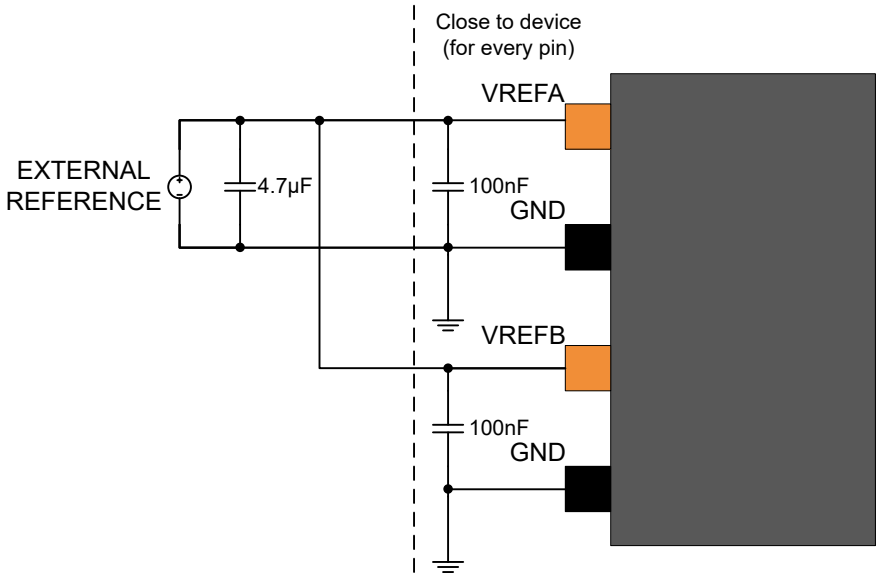


Table 17-2. External Analog Reference Connections

Signal Name	Recommended Pin Connection	Description
VREFx	1.0V to (V_{DDANA} – 0.6V) for ADC Decoupling/filtering capacitors 100 nF(1)(2) and 4.7µF(1)	External reference VREFx for the analog port

.....continued		
Signal Name	**Recommended Pin Connection**	**Description**
GND	—	Ground

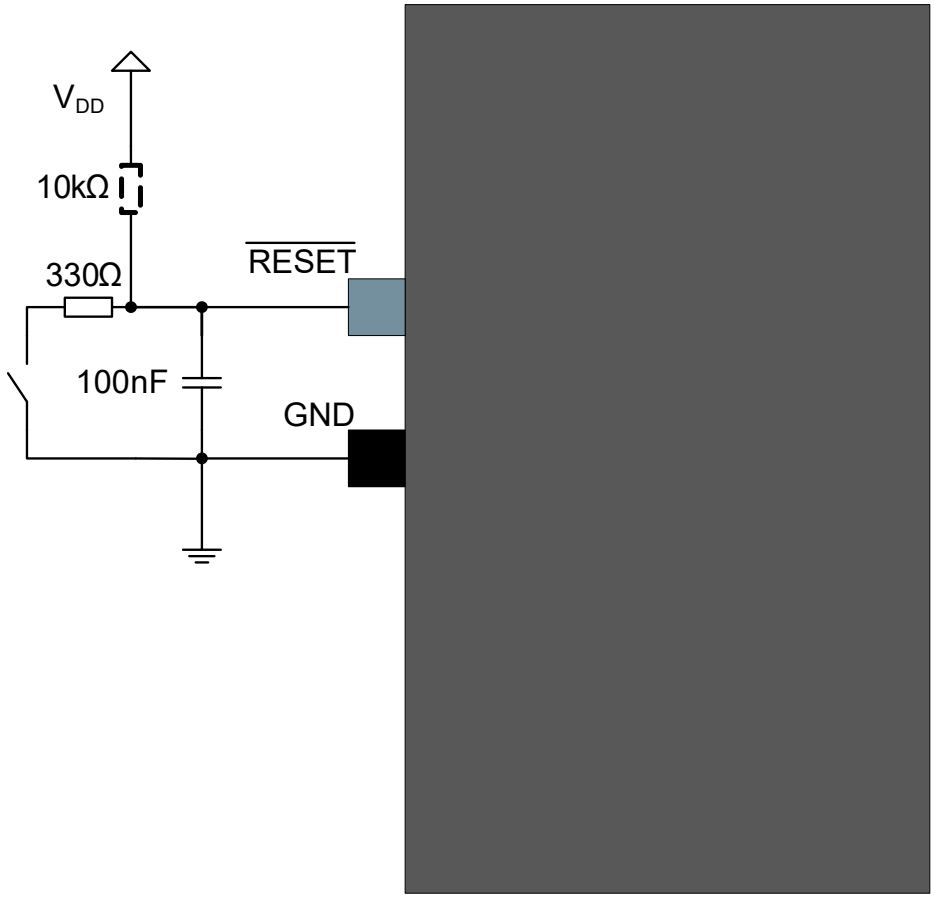
Notes:

1. These values are only given as a typical example.
2. It is recommended that the decoupling capacitor be placed close to the device for each supply pin pair in the signal group.

17.4 External Reset Circuit

The external reset circuit should be connected to the $\overline{\text{RESET}}$ pin when the external reset function is used. If the external reset function has been disabled, the circuit is not necessary. The reset switch can also be removed, if the manual reset is not necessary. The $\overline{\text{RESET}}$ pin itself has an internal pull-up resistor, hence it is optional to add any external pull-up resistor.

Figure 17-5. External Reset Circuit Schematic



A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, i.e. preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Table 17-3. Reset Circuit Connections

Signal Name	Recommended Pin Connection	Description
$\overline{\text{RESET}}$	Reset low level threshold voltage V_{DDIO} = 1.6V - 2.0V: Below 0.33 * V_{DDIO} V_{DDIO} = 2.7V - 3.6V: Below 0.36 * V_{DDIO} Decoupling/filter capacitor 100nF(1) Pull-up resistor 10kΩ(1)(2) Resistor in series with the switch 330Ω(1)	Reset pin

1. These values are only given as a typical example.
2. The SAM R30 features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.

17.5 Unused or Unconnected Pins

For unused pins the default state of the pins will give the lowest current leakage. Thus there is no need to do any configuration of the unused pins in order to lower the power consumption.

17.6 Clocks and Crystal Oscillators

The SAM R30 can be run from internal or external clock sources, or a mix of internal and external sources. An example of usage can be to use the internal 16MHz oscillator as source for the system clock and an external 32.768kHz watch crystal as clock source for the Real-Time counter (RTC).

17.6.1 External Clock Source

Figure 17-6. External Clock Source Schematic

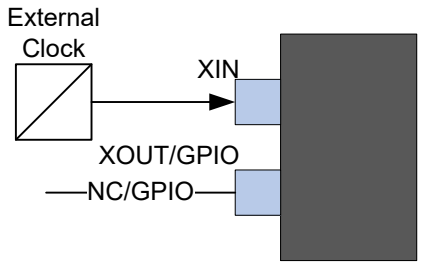
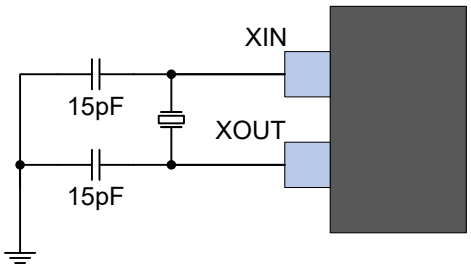


Table 17-4. External Clock Source Connections

Signal Name	Recommended Pin Connection	Description
XIN	XIN is used as input for an external clock signal	Input for inverting oscillator pin
XOUT/GPIO	Can be left unconnected or used as normal GPIO	NC/GPIO

17.6.2 Crystal Oscillator

Figure 17-7. Crystal Oscillator Schematic



The crystal should be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

Table 17-5. Crystal Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN	Load capacitor 15pF(1)(2)	External crystal between 0.4 to 32MHz
XOUT	Load capacitor 15pF(1)(2)	

1. These values are only given as a typical example.
2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

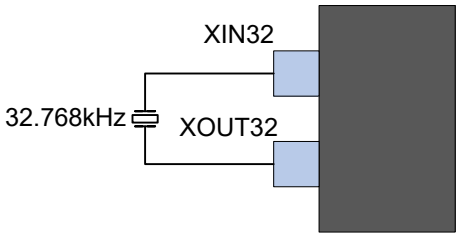
17.6.3 External Real Time Oscillator

The low frequency crystal oscillator is optimized for use with a 32.768kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM R30 oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The typical parasitic load capacitance values are available in the Electrical Characteristics section. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in Figure 17-8.

Figure 17-8. External Real Time Oscillator without Load Capacitor



To improve accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown in Figure 17-9.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

Figure 17-9. External Real Time Oscillator with Load Capacitor

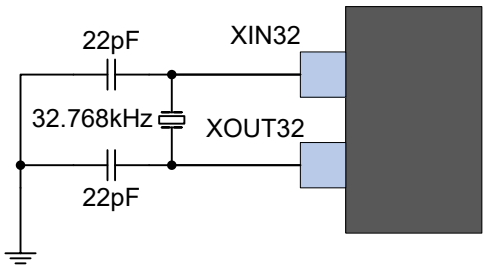


Table 17-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 22pF(1)(2)	Timer oscillator input
XOUT32	Load capacitor 22pF(1)(2)	Timer oscillator output

1. These values are only given as typical examples.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

Note: In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.

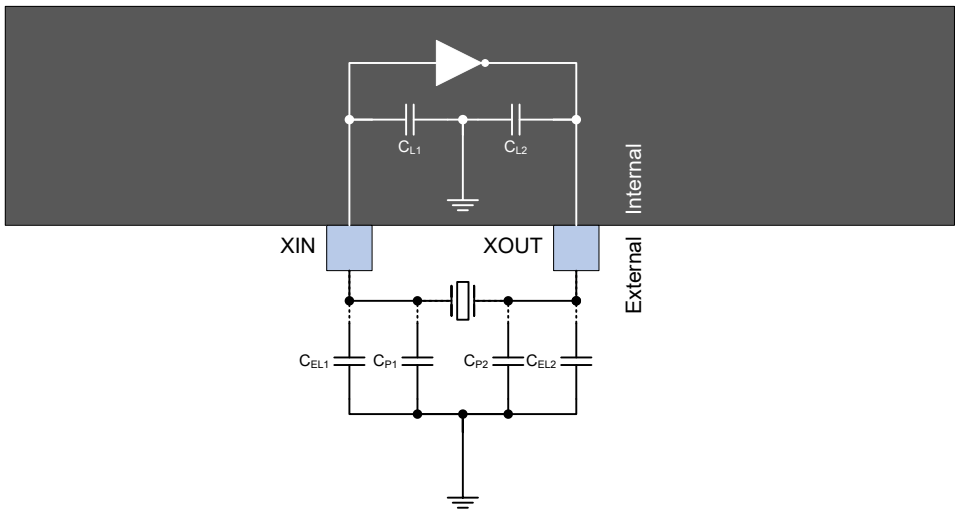
Related Links

7.1 Multiplexed Signals

17.6.4 Calculating the Correct Crystal Decoupling Capacitor

The model shown in Figure 17-10 can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors C_{Ln}, external parasitic capacitance C_{ELn} and external load capacitance C_{Pn}.

Figure 17-10. Crystal Circuit With Internal, External and Parasitic Capacitance



Using this model the total capacitive load for the crystal can be calculated as shown in the equation below:

$$\sum C_{\text{tot}} = \frac{(C_{L1} + C_{P1} + C_{\text{EL1}})(C_{L2} + C_{P2} + C_{\text{EL2}})}{C_{L1} + C_{P1} + C_{\text{EL1}} + C_{L2} + C_{P2} + C_{\text{EL2}}}$$

where C_{tot} is the total load capacitance seen by the crystal. This value should be equal to the load capacitance value found in the crystal manufacturer datasheet.

The parasitic capacitance C_{ELn} can in most applications be disregarded as these are usually very small. If accounted for, these values are dependent on the PCB material and PCB layout.

For some crystal the internal capacitive load provided by the device itself can be enough. To calculate the total load capacitance in this case. C_{ELn} and C_{Pn} are both zero, $C_{L1} = C_{L2} = C_L$, and the equation reduces to the following:

$$\sum C_{\text{tot}} = \frac{C_L}{2}$$

See the related links for equivalent internal pin capacitance values.

Related Links

15.12.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

17.7 Programming and Debug Ports

For programming and/or debugging the SAM R30, the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Microchip and third party programmers and debuggers, like the Atmel-ICE, SAM-ICE or SAM R30 Xplained Pro (SAM R30 evaluation kit) Embedded Debugger.

Refer to the Atmel-ICE, SAM-ICE or SAM R30 Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM R30 Xplained Pro evaluation board supports programming and debugging through the onboard embedded debugger so no external programmer or debugger is needed.

Note: A pull-up resistor on the SWCLK pin is critical for reliable operation. Refer to related link for more information.

Figure 17-11. SWCLK Circuit Connections

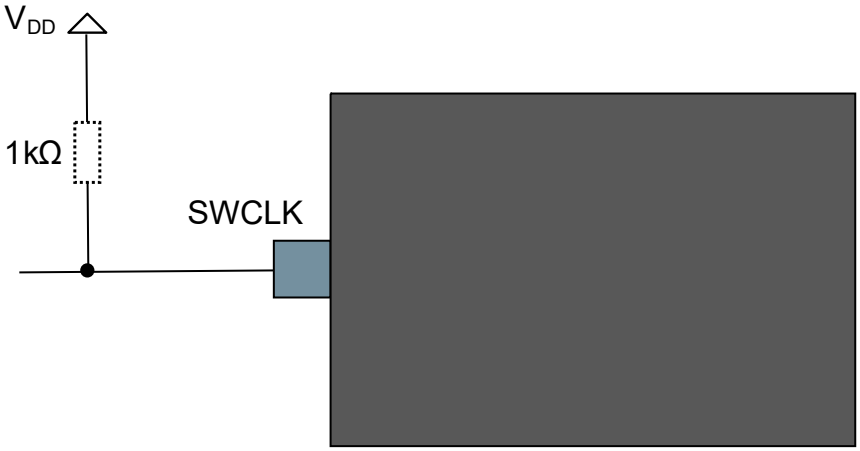


Table 17-7. SWCLK Circuit Connections

Pin Name	Description	Recommended Pin Connection
SWCLK	Serial wire clock pin	Pull-up resistor 1kΩ

Related Links

17.1.1 Operation in Noisy Environment

17.7.1 Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in Figure 17-12 with details described in Table 17-8.

Figure 17-12. Cortex Debug Connector (10-pin)

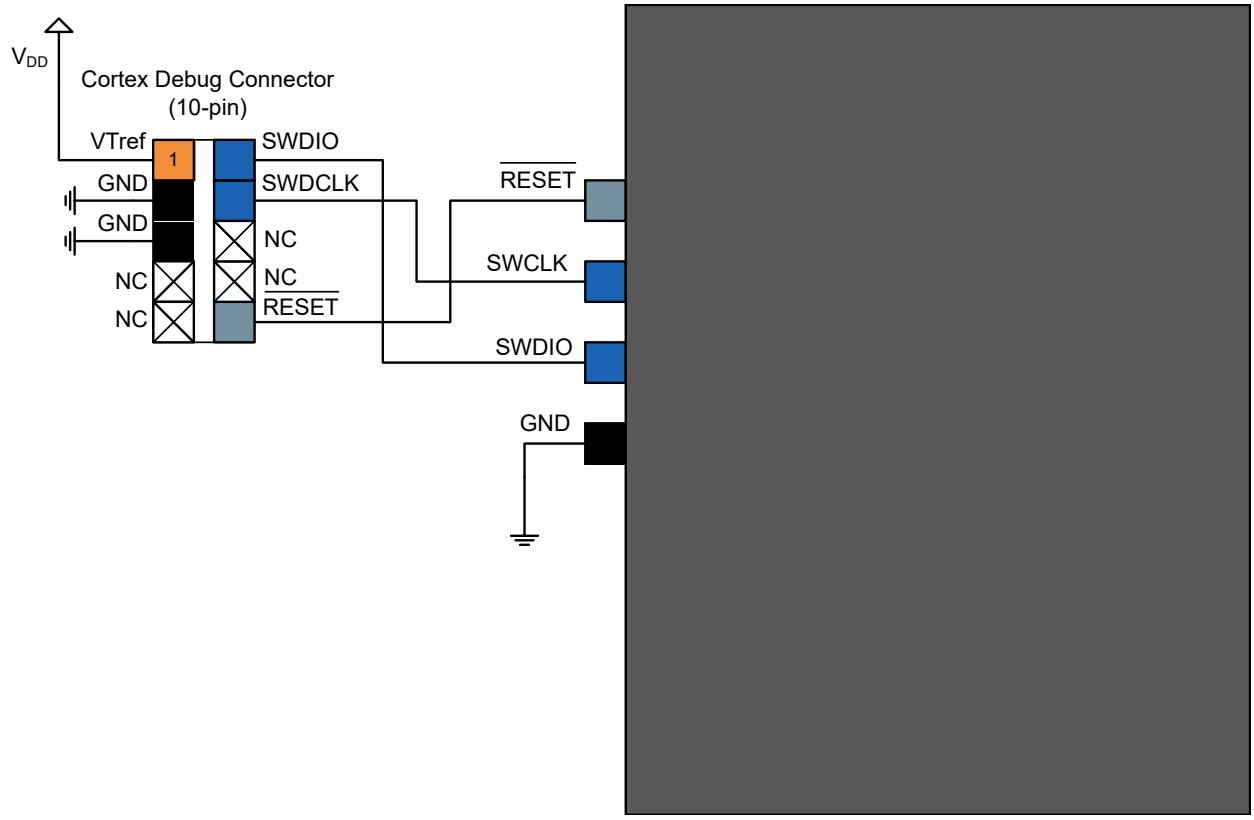


Table 17-8. Cortex Debug Connector (10-pin)

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTref	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

17.7.2 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

The JTAGICE3 debugger and programmer does not support the Cortex Debug Connector (10-pin) directly, hence a special pinout is needed to directly connect the SAM R30 to the JTAGICE3, alternatively one can use the JTAGICE3 squid cable and manually match the signals between the JTAGICE3 and SAM R30. Figure 17-13 describes how to connect a 10-pin header that support connecting the JTAGICE3 directly to the SAM R30 without the need for a squid cable. This can also be used for the Atmel-ICE AVR connector port.

The JTAGICE3 squid cable or the JTACICE3 50mil cable can be used to connect the JTAGICE3 programmer and debugger to the SAM R30. 17.7.2 10-pin JTAGICE3 Compatible Serial Wire Debug Interface illustrates the correct pinout for the JTAGICE3 50 mil, and details are given in Table 17-9.

Figure 17-13. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

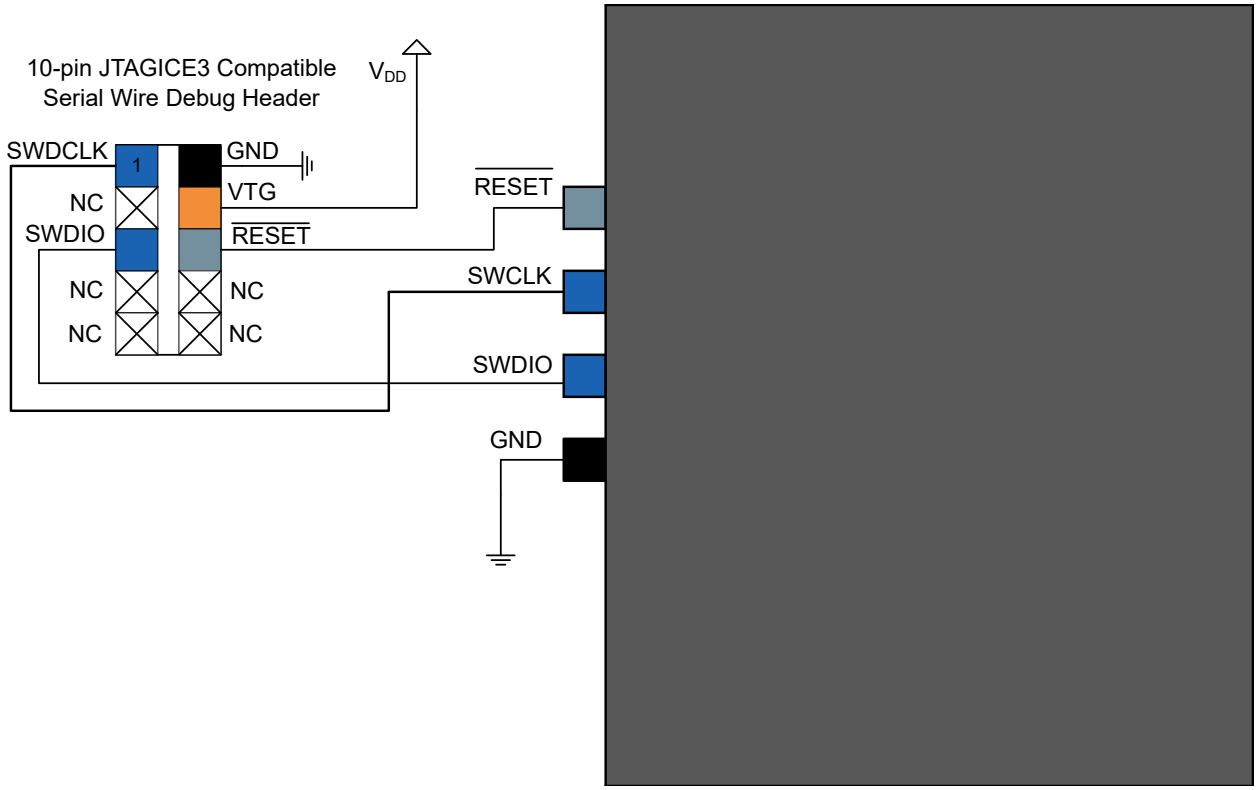


Table 17-9. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTG	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

17.7.3 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g. the SAM-ICE, the signals should be connected as shown in Figure 17-14 with details described in Table 17-10.

Figure 17-14. 20-pin IDC JTAG Connector

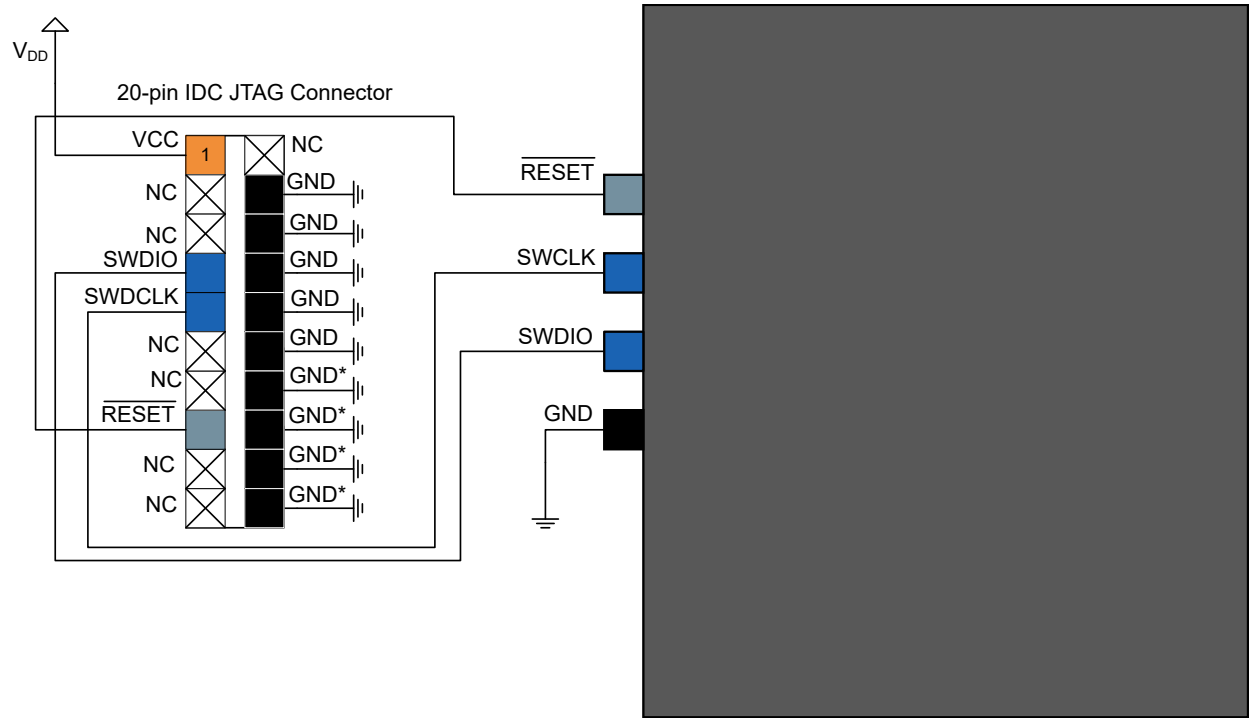


Table 17-10. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device V_{DD}
GND	Ground
GND*	These pins are reserved for firmware extension purposes. They can be left unconnected or connected to GND in normal debug environment. They are not essential for SWD in general.

17.8 USB Interface

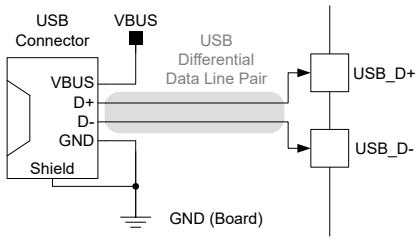
The USB interface consists of a differential data pair (D+/D-) and a power supply (VBUS, GND).

Refer to the Electrical Characteristics section for operating voltages which will allow USB operation.

Table 17-11. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections. • USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • For a tightly coupled differential pair,the signal routing should be as parallel as possible, with a minimum number of angles and vias.	USB full speed / low speed positive data upstream pin
D-		USB full speed / low speed negative data upstream pin

Figure 17-15. Low Cost USB Interface Example Schematic

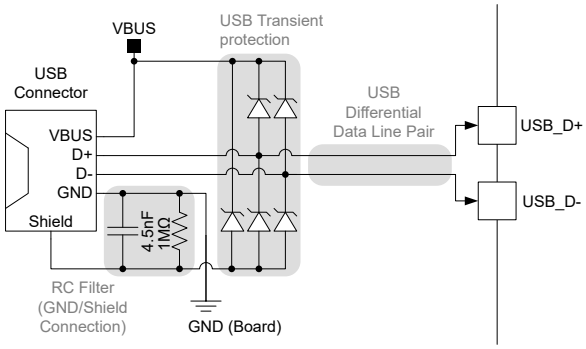


It is recommended to increase ESD protection on the USB D+, D-, and VBUS lines using dedicated transient suppressors. These protections should be located as close as possible to the USB connector to reduce the potential discharge path and reduce discharge propagation within the entire system.

The USB FS cable includes a dedicated shield wire that should be connected to the board with caution. Special attention should be paid to the connection between the board ground plane and the shield from the USB connector and the cable.

Tying the shield directly to ground would create a direct path from the ground plane to the shield, turning the USB cable into an antenna. To limit the USB cable antenna effect, it is recommended to connect the shield and ground through an RC filter.

Figure 17-16. Protected USB Interface Example Schematic



Related Links

15. Electrical Characteristics

18. Design Considerations

18.1 RF Balun

The RF front end for SAMR30 can be designed with a chip balun or with a discrete balun. The reference schematic for both the types of RF frontend is available in this section. Microchip recommends to add two pi-structures in the RF frontend between the balun output and input to the antenna for proper impedance matching. The first pi-structure should be used for matching the impedance from balun to a 50Ω point and the second pi-structure should be used for matching the impedance from the antenna to a 50Ω point. This 50Ω point can be used to place a test point that can be used during production for conducted RF measurements.

Figure 18-1. RF Frontend Section with Chip Balun and Harmonic Filter

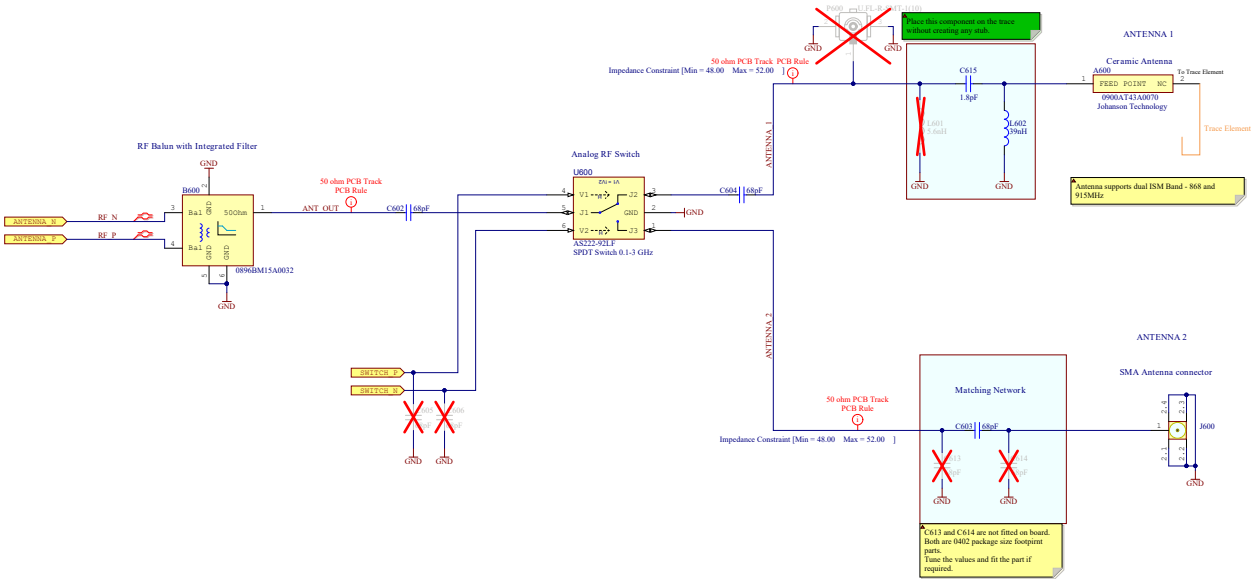
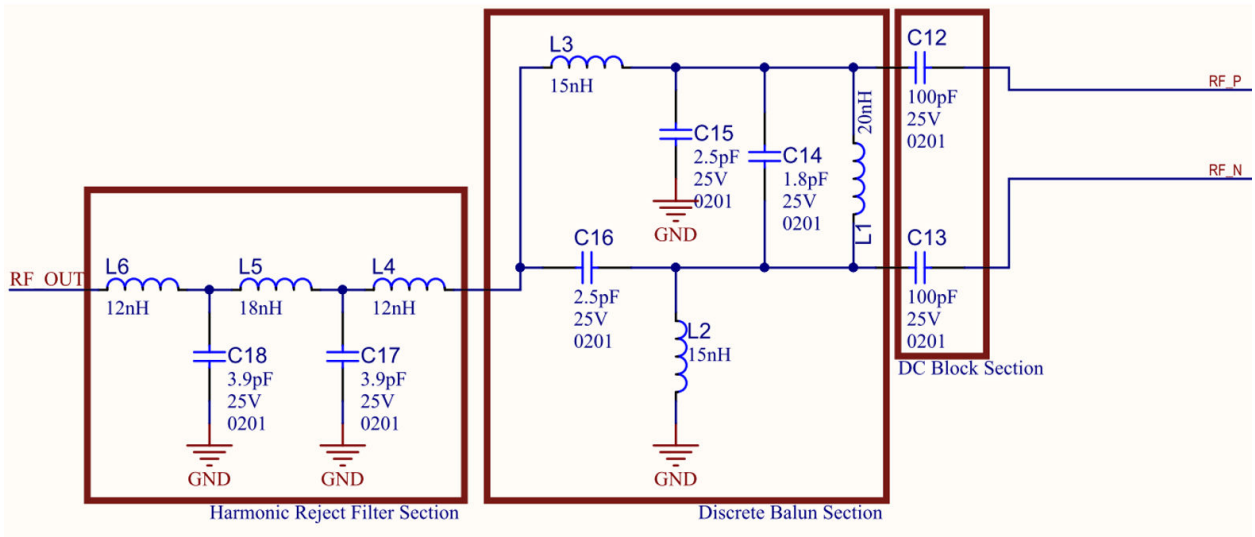


Figure 18-2. RF Frontend Section with Discrete Balun and Harmonic Filter



18.2 Placement and Routing Guidelines

It is critical to follow the recommendations listed in this section to achieve the best RF performance.

18.2.1 Power and Ground

The following are the recommendations for a better design with respect to power and ground.

- Dedicate one layer as a ground plane with a copper polygon pour running throughout the plane. Make sure this ground plane is not broken up by any signal traces. This ground layer must be selected to be immediately below the layer containing the RF traces routed from the ATSAMR30 IC to the antenna.
- Place sufficient ground vias throughout the board to ensure the inductance to the ground path is the least possible.
- It is recommended that all power supply traces have heavy copper fill planes to ensure the lowest possible inductance.
- It is recommended that all decoupling capacitors have a dedicated ground via placed next to the capacitor pad. These capacitors must be placed as close as possible to the IC pin that it is filtering.

Figure 18-3. Correct Routing through Capacitor Pad

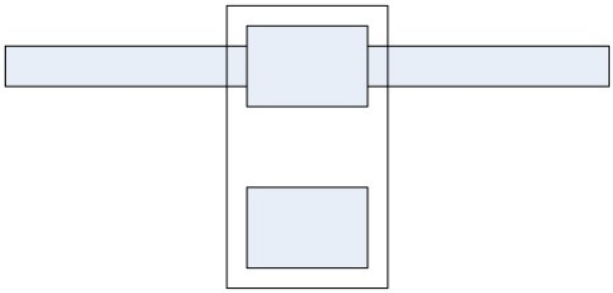
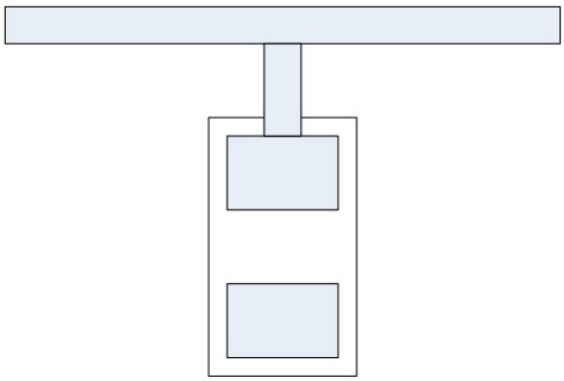
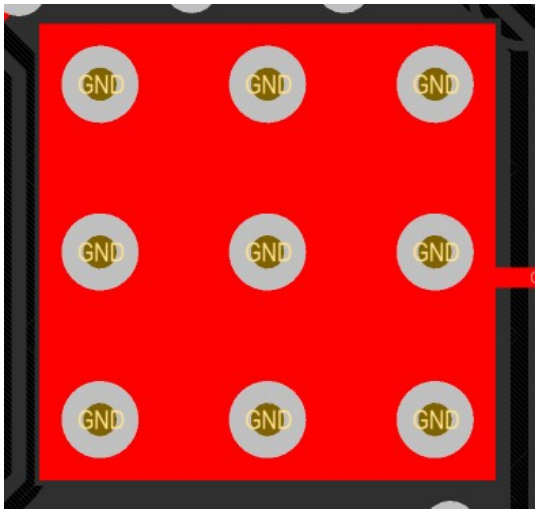


Figure 18-4. Incorrect Stub Routing to Capacitor Pad



- Wherever possible, avoid using the switched mode power supply to source the SAMR30 as the noise induced in the power rail can impact the RF performance. Microchip recommends using LDO for sourcing the SAMR30. If this is unavoidable, add a sufficient noise filter circuit to ensure the power rail is clean from high frequency noise ripples.
- The center ground paddle of the ATSAMR30 device must be solidly interfaced to the ground plane through a grid of 3x3 ground vias, as shown in the following figure. It is recommended that the minimum via be 0.3 mm hole size and 0.75 mm drill size.

Figure 18-5. Proper Grounding of the Center Ground Paddle

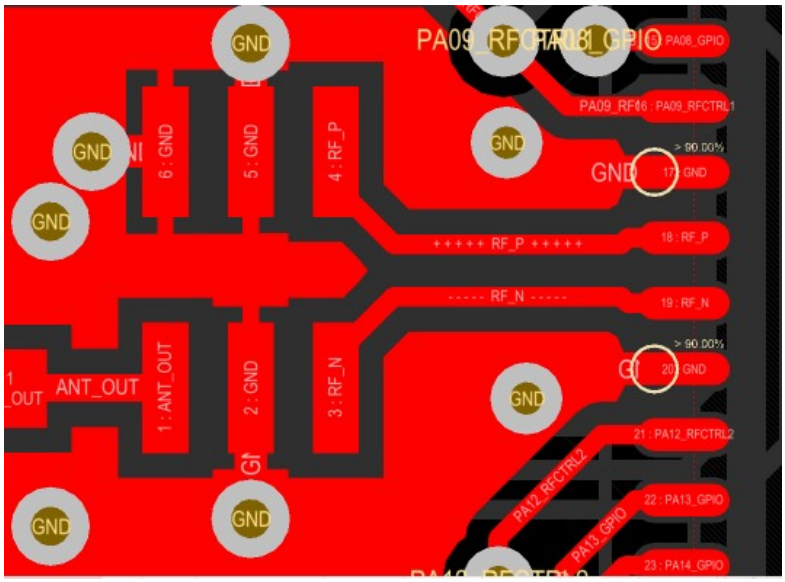


18.2.2 RF Traces and Components

The following are the recommendations for a better design with respect to RF traces and components in the RF frontend.

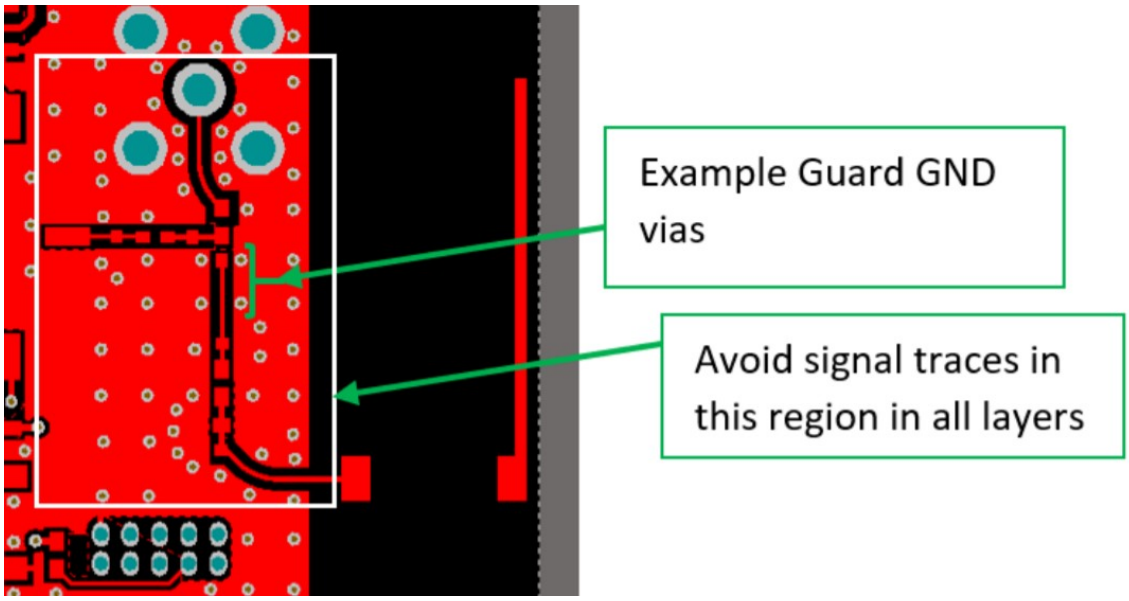
- The RF traces routed from RFP and RFN of the ATSAMR30 must be designed for a 100Ω differential impedance. These traces routed to the balun must be kept as short as possible to reduce path loss and avoid opportunity for the trace to pick up noise.

Figure 18-6. 100Ω Differential Trace from RFP and RFN to Chip Balun



- Place guard ground vias along the RF trace. The PCB layer directly below the RF trace must have a ground polygon pour.
- Do not have any signal traces below/adjacent to the RF trace.
- Do not use thermal relief pads for the ground pads of all components in the RF frontend. These pads must be completely flooded with ground copper polygon pour. Place dedicated vias to ground for all these individual components.

Figure 18-7. Reference Placement of Guard Ground Vias and Complete Copper Flooding of Ground Pads of Components in the RF Frontend



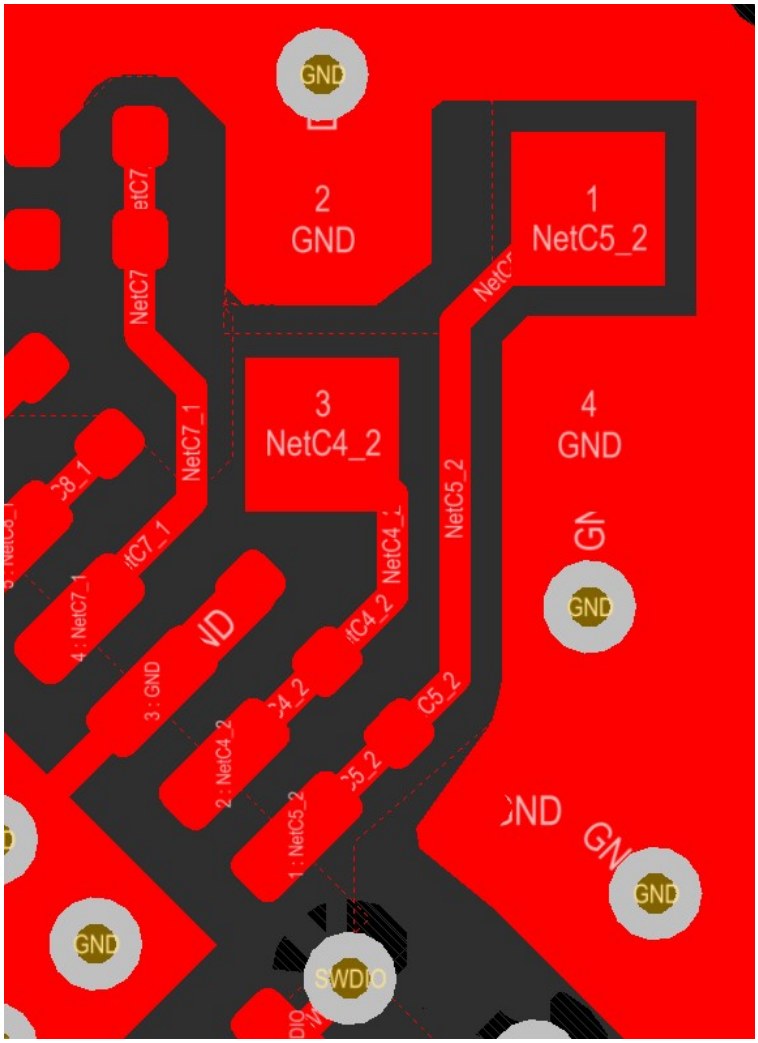
- Keep any components that may radiate noise or signals within the 850-950 MHz frequency band away from the antenna and the RF traces and, if possible, shield these components. Any noise radiated from the board in this frequency band degrades the RF performance of the module.

18.2.3 Crystal Routing

The following are the recommendations for a better design with respect to trace routing to crystal.

- Keep the traces from the IC pad to the crystal as short as possible.
- Place guard ground vias all along the trace routed to crystal.
- Surround the area around the crystal with ground polygon pour and place sufficient ground vias in this polygon pour.

Figure 18-8. Reference Routing of Trace to Crystal



18.3 Antenna

- Make sure to choose an antenna that covers the desired frequency band, in other words, 868 MHz (European), 902-928 MHz (North American), 779 to 78 MHz (Chinese) and 915 to 930 MHz (Japanese).
- Ensure the PCB trace running from the balun output to antenna is designed for single-ended 50Ω impedance.
- Follow the footprint and other recommendations from the antenna data sheet.
- Ensure the dimension of the ground plane in the board meets the dimension requirement specified in the antenna data sheet. A reduced ground plane dimension reduces the efficiency achieved with the same antenna.
- Place the antenna impedance-matching components as close to the antenna pad as possible.

18.4 Calibrating 16 MHz Crystal Oscillating Frequency in Prototype

In prototype boards, tune the oscillating frequency of the 16 MHz crystal to the desired frequency using the external load capacitors and when the XTAL_TRIM bits are with a value of 0x7. The user must tune the board with XTAL_TRIM as 0x7 to ensure that during production calibration, there is a scope to swing either side of the desired frequency to achieve the desired frequency tolerance. XTAL_TRIM bits of XOSC_CTRL register (AT86RF212B) controls the internal trimming capacitor value. Typically, in production, the frequency calibration is performed by measuring the frequency variation from the desired value and compensating for this offset by adjusting the XTAL_TRIM bits. Thereby, the internal trimming capacitance is either increased or decreased. The specific value of

XTAL_TRIM where the frequency variation is within the limits set in production is stored in NVM. This value is loaded into XTAL_TRIM during boot-up of the firmware through the initialization routine. The limit in production can be set at +/-5 kHz for achieving the best possible frequency tolerance.

Note: The choice of crystal also has an impact on the frequency tolerance.

In order to configure XTAL_TRIM with value 0x7 for the calibration, modify the method `tfa_continuous_tx_start()` in the `tfa.c` file available in the Performance analyzer ASF example project. The following is the code snippet to be inserted in the `tfa.c` file.

```
void tfa_continuous_tx_start(continuous_tx_mode_t tx_mode, bool random_content)
{
    uint8_t txcwdata[128];
    uint8_t i;

    /* step 3,6: Channel is assumed to be set before */
    trx_reg_write(RG_TRX_STATE, CMD_TRX_OFF);
             trx_bit_write(SR_XTAL_TRIM,0x7); //Write XTAL_TRIM bits with a value 0x7
    /* step 7: Enable continuous transmission - step #1 */
    trx_reg_write(0x36, 0x0F);
```

This test can be done by generating a CW tone at a specific channel and measuring the frequency tolerance with a Spectrum analyzer. The following are the steps that must be performed during this calibration process.

1. Load the SAMR30 device with Performance analyzer firmware. Refer to the certification user guide document on the prerequisites, installation and procedure to configure the SAMR30 in CW mode. Note that with the default FW, the CW tone will be generated at Fc -0.1 MHz. Refer to section 21, Continuous Transmission Test Mode of the SAMR30 Data Sheet, for more details.
2. Follow the steps in section General Configuration and Continuous TX – CW Mode under the section Configuring SAMR30 in Different Test Modes to initiate a CW tone at a desired frequency. For reference, let the configuration be Channel 1, i.e., 906 MHz.
3. Measure the frequency of the peak output with a spectrum analyzer. The difference between the measured value and 905.9 MHz, i.e., 906 – 0.1 = 905.9 MHz, is the frequency offset as observed in the board. This frequency offset could be adjusted to be as close as possible to the desired frequency by adjusting the external load capacitors added for 16 MHz crystal.
4. Adjust the external capacitor to a nearby value and verify the frequency tolerance. Repeat the step until the lowest possible frequency tolerance is achieved.
5. Verify the frequency tolerance in additional samples with the external load capacitor as identified in step 4.
6. Freeze the BoM for the external load capacitor based on verification done in the above steps. It is recommended to perform the XTAL calibration in production and to arrive at the XTAL_TRIM value for each unit that results in an accurate clock source for RF.

The same steps can be done by routing the CLKM from AT86RF21B to the GPIO pins of ATSAMR30 through the GCLK I/O peripheral function. This allows the measurement to be done in < 16 MHz using a frequency counter.

19. Conventions

19.1 Numerical Notation

Table 19-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous.
0x3B24	Hexadecimal number
X	Represents an unknown or don't care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

19.2 Memory Size and Type

Table 19-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

19.3 Frequency and Time

Symbol	Description
kHz	1kHz = 10^3Hz = 1,000Hz
KHz	1KHz = 1,024Hz, 32KHz = 32,768Hz
MHz	10^6 = 1,000,000Hz
GHz	10^9 = 1,000,000,000Hz
s	second
ms	millisecond

.....continued	
Symbol	**Description**
µs	microsecond
ns	nanosecond

19.4 Registers and Bits

Table 19-3. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read.
PERIPHERAL*i*	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a power reset. This is also the value of registers in a peripheral after performing a software reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a one to a bit in the CLR register will clear the corresponding bit in both registers, while writing a one to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

20. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 20-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	AMBA Advanced High-performance Bus
AMBA®	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog reference voltage
BLB	Boot Lock Bit
BOD	Brown-out detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt flag
INT	Interrupt
MBIST	Memory built-in self-test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer

.....continued

Abbreviation	Description
NMI	Non-maskable interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Non-Volatile Memory
NVMCTRL	Non-Volatile Memory Controller
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period
PM	Power Manager
POR	Power-on reset
PORT	I/O Pin Controller
PTC	Peripheral Touch Controller
PWM	Pulse Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SERCOM	Serial Communication Interface
SMBus™	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SUPC	Supply Controller
SWD	Serial Wire Debug
TC	Timer/Counter
TCC	Timer/Counter for Control Applications
TRNG	True Random Number Generator
TX	Transmitter/Transmit
ULP	Ultra-low power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
USB	Universal Serial Bus
V_{DD}	Common voltage to be applied to VDDIO, VDDIN and VDDANA
V_{DDIN}	Digital supply voltage
V_{DDIO}	Digital supply voltage
V_{DDANA}	Analog supply voltage

.....continued

Abbreviation	Description
VREF	Voltage reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

21. Continuous Transmission Test Mode

21.1 Overview

The AT86RF212B offers a Continuous Transmission Test Mode to support application and production tests as well as certification tests. Using this test mode, the radio transceiver transmits continuously a previously transferred frame (PRBS mode) or a continuous wave signal (CW mode).

The AT86RF212B uses I/Q modulation for both, PRBS mode and CW mode. In CW mode, this results in a signal which is not placed at the selected channel center frequency F_C, but at 0.1 or 0.25MHz apart this frequency. One out of four different signal frequencies per channel can be transmitted:

- $f_1 = F_C + 0.25MHz$ using O-QPSK 1000kb/s mode
- $f_2 = F_C - 0.25MHz$ using O-QPSK 1000kb/s mode
- $f_3 = F_C + 0.1MHz$ using O-QPSK 400kb/s mode
- $f_4 = F_C - 0.1MHz$ using O-QPSK 400kb/s mode

As a side effect of I/Q modulation, CW mode shows some unwanted signal components based on finite image rejection and non-linearities.

In addition to the above mentioned modes – a CW mode which directly uses the PLL signal without I/Q modulation. This is the recommended mode because the signal is placed at the selected channel center frequency F_C and unwanted signal components are significantly lower.

PRBS mode requires data in the frame buffer, that is a valid PHR followed by PSDU data. After transmission of two non-PSDU octets, PSDU data is repeated continuously.

Related Links

14.3.1 Introduction – IEEE 802.15.4-2006 Frame Format
14.4.8.2 RF Channel Selection

21.2 Configuration

Detailed programming sequences for PRBS, CW and additional CW mode are showed in the tables below. The column R/W informs about writing (W) or reading (R) a register or the Frame Buffer.

Table 21-1. PRBS and CW Mode Programming Sequence

Step	Action	Register	R/W	Value	Description
1	RESET				Reset AT86RF212B
2	Register access	0x0E	W	0x01	Set IRQ mask register, enable IRQ_0 (PLL_LOCK)
3	Register access	0x02	W	0x03	Set radio transceiver state TRX_OFF
4	Register access		W		Set channel
5	Register access		W		Set TX output power. For CW mode, GC_TX_OFFS should be set to three(1).
6	Register access	0x01	R	0x08	Verify TRX_OFF state
7	Register access	0x36	W	0x0F	

.....continued

Step	Action	Register	R/W	Value	Description
8	Register access	0x0C	W	0x00 0x04 0x08 0x0C 0x1C 0x0A 0x0E	Select PRBS mode with modulation scheme or CW mode with carrier position: PRBS mode, BPSK-20 PRBS mode, BPSK-40 PRBS mode, OQPSK-SIN-RC-100 PRBS mode, OQPSK-SIN-250 PRBS mode, OQPSK-RC-250 CW mode, CW at Fc - 0.1MHz or CW at Fc + 0.1MHz, see step 9 CW mode, CW at Fc - 0.25MHz or CW at Fc + 0.25MHz, see step 9
9	Frame Buffer write access		W	{PHR, PSDU} {0x01,0x00}{0x01, 0xFF}{0x01, 0x00} {0x01, 0xFF}	PRBS mode: Write PHR value (0x01 … 0x7F) followed by PSDU data. PHR determines how many bytes of the PSDU data are repeated continuously. CW mode, CW at Fc - 0.1MHz CW mode, CW at Fc + 0.1MHz CW mode, CW at Fc - 0.25MHz CW mode, CW at Fc + 0.25MHz
10	Register access	0x1C	W	0x54	
11	Register access	0x1C	W	0x46	
12	Register access	0x02	W	0x09	Enable PLL_ON state
13	Interrupt event	0x0F	R	0x01	Wait for IRQ_0 (PLL_LOCK)
14	Register access	0x02	W	0x02	Initiate transmission, enter BUSY_TX state
15	Measurement				Perform measurement
16	Register access	0x1C	W	0x00	Disable Continuous Transmission Test Mode
17	Reset				Reset AT86RF212B

Table 21-2. Additional CW Mode Programming Sequence

Step	Action	Register	R/W	Value	Description
1	Reset				Reset AT86RF212B rev. C
2	Register access	0x0E	W	0x01	Set IRQ mask register, enable IRQ_0 (PLL_LOCK)
3	Register access	0x02	W	0x03	Set radio transceiver state TRX_OFF
4	Register access		W		Set channel
5	Register access		W		Set TX output power. For CW mode, GC_TX_OFFS should be set to three(1).
6	Register access	0x01	R	0x08	Verify TRX_OFF state

.....continued					
Step	**Action**	**Register**	**R/W**	**Value**	**Description**
7	Register access	0x36	W	0x0F	
8	Register access	0x1C	W	0x54	
9	Register access	0x1C	W	0x42	
10	Register access	0x34	W	0x00	
11	Register access	0x3F	W	0x08	
12	Register access	0x02	W	0x09	Enable PLL_ON state
13	Interrupt event	0x0F	R	0x01	Wait for IRQ_0 (PLL_LOCK)
14	Register access	0x02	W	0x02	Initiate transmission, enter BUSY_TX state
15	Measurement				Perform measurement
16	Register access	0x1C	W	0x00	Disable Continuous Transmission Test Mode
17	Reset				Reset AT86RF212B rev. C

1. Changing the output power during continuous transmission is not allowed

22. Errata

The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

22.1 DSU

22.1.1 Reference:16144

When the device is waking from standby retention mode, selected alternate function on PA30 (SERCOM for example) will be lost, and it functions as SWCLK pin and can switch the device to debug mode.

Workaround
Disable the debugger Hot-Plugging detection by setting the security bit. Security is set by issuing the NVMCTRL SSB command.

Affected Silicon Revisions

C							
X							

22.2 DFLL48M

22.2.1 Reference:9905

The DFLL clock must be requested before being configured, otherwise, a write access to a DFLL register can freeze the device.

Workaround
Write a ‘0’ to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

Affected Silicon Revisions

C							
X							

22.2.2 Reference:16192

If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out-of-bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might, therefore, be false out of bounds interrupts.

Workaround
Check that the lockbits: DFLLLCKC and DFLLLCKF in the OSCCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

Affected Silicon Revisions

C							
X							

22.2.3 Reference:16193

The DFLL status bits in the STATUS register during the USB clock recovery mode can be wrong after a USB suspend state.

Workaround
Do not monitor the DFLL status bits in the STATUS register during the USB clock recovery mode.

Affected Silicon Revisions

C							
X							

22.3 DMAC

22.3.1 Reference:15670

When using more than one DMA channel, and if one of these DMA channels has a linked descriptor, a fetch error can appear on this channel.

Workaround
Do not use linked descriptors, make a software link instead:

Replace the channel that used the linked descriptor with two-channel DMA (with the linked descriptor disabled) handled by the two channels event system:

- DMA channel 0 transfer completion is able to send a conditional event for DMA channel 1 (via event system with configuration of BTCTRL.EVOSEL = BLOCK for channel 0 and configuration CHCTRLB.EVACT = CBLOCK for channel 1).
- On the transfer complete reception of DMA channel 0, immediately re-enable channel 0.
- Then, DMA channel 1 transfer completion is able to send a conditional event for DMA channel 0 (via event system with configuration of BTCTRL.EVOSEL = BLOCK for channel 1 and configuration CHCTRLB.EVACT = CBLOCK for channel 0).
- On the transfer complete reception of DMA channel 1, immediately re-enable the channel 1.
- The mechanism can be launched by sending a software event on DMA channel 0.

Affected Silicon Revisions

C							
X							

22.3.2 Reference:15683

When at least one channel using linked descriptors is already active, enabling another DMA channel (with or without linked descriptors) can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch.

Workaround
This happens if the channel number of the channel being enabled is lower than the channel already active.

When enabling a DMA channel while other channels using linked descriptors are already active, the channel number of the new channel enabled must be greater than the other channel numbers.

Affected Silicon Revisions

C							

X							

22.4 FDPLL

22.4.1 Reference:15753

When the FDPLL ratio value in the DPLLRATIO register is changed on the fly, STATUS.DPLLLDRTO will not be set even though the ratio is updated.

Workaround
Monitor the INTFLAG.DPLLLDRTO instead of STATUS.DPLLLDRTO to get the status for the DPLLRATIO update.

Affected Silicon Revisions

C							
X							

22.5 PORT

22.5.1 Reference:15611

PORT read/write attempts on non-implemented registers, including addresses beyond the last implemented register group (PA, PB,...) do not generate a PAC protection error.

Workaround
None

Affected Silicon Revisions

C							
X							

22.6 EIC

22.6.1 Reference:14417

Access to EIC_ASYNCH register in 8/16-bit mode is not functional.

Workaround

- Writing in 8-bit mode will also write this byte in all bytes of the 32-bit word.
- Writing the higher 16 bits will also write the lower 16 bits.
- Writing the lower 16 bits will also write the higher 16 bits.

Two workarounds are available:

- Use 32-bit write mode.
- Write only the lower 16 bits. (This will write the upper 16 bits also, but does not impact the application.)

Affected Silicon Revisions

C							

X							

22.6.2 Reference:15278

When the EIC is configured to generate an interrupt on a low level or rising edge or both edges (CONFIGn.SENSEx) with the filter enabled (CONFIGn.FILTENx), a spurious flag might appear for the dedicated pin on the INTFLAG.EXTINT[x] register as soon as the EIC is enabled using CTRLA ENABLE bit.

Workaround
Clear the INTFLAG bit once the EIC enabled and before enabling the interrupts.

Affected Silicon Revisions

C							
X							

22.6.3 Reference:15279

Changing the NMI configuration (CONFIGn.SENSEx) on the fly may lead to a false NMI interrupt.

Workaround
Clear the NMIFLAG bit once the NMI is modified.

Affected Silicon Revisions

C							
X							

22.6.4 Reference:16103

When the asynchronous edge detection is enabled and the system is in standby mode, only the first edge will generate an event. The following edges will not generate events until the system wakes up.

Workaround
Asynchronous edge detection does not work, instead, use the synchronous edge detection (ASYNCH.ASYNCH[x] = 0). In order to reduce power consumption when using synchronous edge detection, either set the GCLK_EIC frequency as low as possible or select the ULP32K clock (EIC CTRLA.CKSEL = 1).

Affected Silicon Revisions

C							
X							

22.7 Device

22.7.1 Reference:15581

On pin PA24 and PA25, the pull-up and pull-down configuration is not disabled automatically when alternative pin function is enabled.

Workaround
For pin PA24 and PA25, the GPIO pull-up and pull-down must be disabled before enabling alternative functions on them.

Affected Silicon Revisions

C							
X							

22.7.2 Reference:16225

When configured in HS or FastMode+, SDA and SCL fall times are shorter than the I^2C specification requirement and can lead to reflection.

Workaround
When reflection is observed, a 100 ohms serial resistor can be added on the impacted line.

Affected Silicon Revisions

C							
X							

22.8 ADC

22.8.1 Reference:14431

The LSB of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution.

Workaround
Use the 12-bit resolution and take only at least 8 bits or 10 bits, if necessary.

Affected Silicon Revisions

C							
X							

22.8.2 Reference:15463

In standby sleep mode, when the ADC is in free-running mode (CTRLC.FREERUN = 1) and the RUNSTDBY bit is set to 0 (CTRLA.RUNSTDBY = 0), the ADC keeps requesting its generic clock.

Workaround
Stop the free-running mode (CTRLC.FREERUN = 0) before entering standby sleep mode

Affected Silicon Revisions

C							
X							

22.8.3 Reference:16027

ADC SYNCBUSY.SWTRIG gets stuck to ‘1’ after wake-up from standby sleep mode.

Workaround
Ignore ADC SYNCBUSY.SWTRIG status when waking up from standby sleep mode. The ADC result can be read after INTFLAG.RESRDY is set. To start the next conversion, write ‘1’ to SWTRIG.START.

Affected Silicon Revisions

C							
X							

22.9 TC

22.9.1 Reference:15056

When clearing the STATUS.PERBUFV/STATUS.CCBUFx flag, SYNCBUSY flag is released before the PERBUF/CCBUFx register is restored to its appropriate value.

Workaround
Clear successively twice the STATUS.PERBUFV/STATUS.CCBUFx flag to ensure that the PERBUF/CCBUFx register value is properly restored before updating it.

Affected Silicon Revisions

C							
X							

22.10 TCC

22.10.1 Reference:14817

Advance capture mode (CAPTMIN CAPTMAX LOCMIN LOCMAX DERIV0) does not work if an upper channel is not in one of these modes. Example: when CC[0] = CAPTMIN, CC[1] = CAPTMAX, CC[2] = CAPTEN and CC[3] = CAPTEN, CAPTMIN and CAPTMAX will not work.

Workaround
Basic capture mode must be set in a lower channel and advance capture mode in an upper channel.

Example: CC[0] = CAPTEN , CC[1] = CAPTEN , CC[2] = CAPTMIN, CC[3] = CAPTMAX

All capture will be done as expected.

Affected Silicon Revisions

C							
X							

22.10.2 Reference:15057

When clearing STATUS.xxBUFV flag, SYNCBUSY is released before the register is restored to its appropriate value.

Workaround
To ensure that the register value is properly restored before updating this same register through xx or xxBUF with a new value, the STATUS.xxBUFV flag must be cleared successively two times.

Affected Silicon Revisions

C							
X							

22.10.3 Reference:15625

Using TCC in dithering mode with external retrigger events can lead to an unexpected stretch of right-aligned pulses or shrink of left-aligned pulses.

Workaround

Do not use retrigger events/actions when TCC is configured in dithering mode.

Affected Silicon Revisions

C							
X							

22.11 EVSYS

22.11.1 Reference:14532

Using synchronous, spurious overrun can appear with a generic clock for the channel always on.

Workaround

- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to ‘1’.
- No penalty is introduced.

Affected Silicon Revisions

C							
X							

22.11.2 Reference:14835

The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag.

Workaround

For applications using event generators other than the software event, monitor the OVR flag.

For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

Affected Silicon Revisions

C							
X							

22.12 SERCOM

22.12.1 Reference:13852

In USART autobaud mode, missing stop bits are not recognized as inconsistent sync (ISF) or framing (FERR) errors.

Workaround

None

Affected Silicon Revisions

C							
X							

23. References

[1]	IEEE Standard 802.15.4™-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
[2]	IEEE Standard 802.15.4™-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
[3]	IEEE Standard 802.15.4c™-2009: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): Amendment 2: Alternative Physical Layer Extension to support one or more of the Chinese 314-316MHz, 430-434MHz, and 779-787MHz bands.
[4]	IEEE Standard 802.15.4™-2011: Low-Rate Wireless Personal Area Networks (WPANs).
[5]	FCC Title 47 (Telecommunication) of the Code of Federal Regulations, Part 15 (Radio Frequency Devices), October 2009.
[6]	ETSI EN 300 220-1 V2.3.1 (2009-04): Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25MHz to 1000MHz frequency range with power levels ranging up to 500mW; Part 1: Technical characteristics and test methods.
[7]	ERC Recommendation 70-03 relating to the use of short range devices (SRD). Version of 18 February 2009.
[8]	ANSI/ESD STM5.1 – 2007, Electrostatic Discharge Sensitivity Testing – Human Body Model (HBM); JESD22-A114E – 2006; CEI/IEC 60749-26 – 2006; AEC-Q100-002-Ref-D.
[9]	ESD-STM5.3.1-1999: ESD Association Standard Test Method for electrostatic discharge sensitivity testing – Charged Device Model (CDM).
[10]	NIST FIPS PUB 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/NIST, November 26, 2001.
[11]	AT86RF212B Software Programming Model.

24. Document Revision History

24.1 Rev. C - 03/2021

Section	Description
Features	• Updated the description of memories • Updated the number of external interrupts as 16 • Updated the number of DMAC channels as 16 • Updated the number of TCC • Updated the speed of ADC as 1 MSPS • Removed programmable gain stage information of ADC • Updated the capacitive touch channel as 18 • Added support for wake-up on touch in standby mode for PTC • Updated the peripherals information
Description	• Updated the number of PTC supported buttons as 18 • Updated the number of software selectable sleep modes as three • Removed support for OFF state
Configuration Summary	• Updated the number of instances for SAM R30E TC as 3 • Updated the number of external interrupt lines for SAM R30G as 16 • Updated the channels of PTC as 6x3 for SAM R30G and 4x3 for SAM R30E • Added PTC channels for self-capacitance • Removed the support for OSC8M • Added the support for OSC16M
MCU Block Diagram	Removed VREFA signal for ADC
Multiplexed Signals	• Updated the peripheral signals of REF and TCC in *Port Function Multiplexing* table • Updated the supply for SAM R30E I/O pin PB22 and PB23 as VDDIN
SERCOM I^2C Pins	Removed PA22 and PA23 pins support for SAMR30E
Signal Description	Updated PORT signal details
Power Domain Overview	Removed DAC, OPAMP, and TRNG from the block diagram
Power Supplies	Removed support for DAC and OPAMP
NVM Temperature Log Row	Added new section
Programming	Updated the reference for POR threshold
DSU	Updated the following registers: • ADDR • DID

.....continued

Section	Description
Clock Distribution	• Updated the figure • Added information on how to customize the clock distribution
Clocks after Reset	Updated the section
GCLK	Updated the following register: • SYNCBUSY • PCHCTRLm
MCLK	Added BUP details
Sleep Mode Operation	Editorial updates
PM - Features	• Removed support for OFF mode • Updated the performance levels as PL0 and PL2
Power Domains	Added *Power Domain Partitioning* figure
PD1	Removed support for AES and TRNG peripheral
PDTOP	Added new section
Sleep Mode Controller	• In *Sleep Mode Entry and Exit Table* updated Mode Entry for IDLE mode and removed OFF mode details • Removed OFF mode information from *Sleep Mode Overview* table
BACKUP Mode	Updated the title name
OFF Mode	Removed this section
Performance Level	Updated *Sleep Modes and Performance Level Transitions* figure
Power Domain Controller	Removed OFF mode details in *Sleep Mode versus Power Domain State Overview* table
Regulators, RAMs, and NVM State in Sleep Mode	Removed OFF mode details in *Regulators, RAMs, and NVM state in Sleep Mode* table
PM	Updated SLEEPCFG register
OSCCTRL	Updated DPLLCTRLB register
32KHz External Crystal Oscillator (XOSC32K) Operation	Updated XOSC32KCTRL as XOSC32K
32KHz Internal Oscillator (OSC32K) Operation	Updated OSC32KCTRL as OSC32K
OSC32KCTRL	Updated OSCULP32K register
Brown-Out Detectors	Updated BODVDD as BOD33
RTC - Features	Updated the number of number of general purpose registers from 4 as 2
32-Bit Counter (Mode 0)	Editorial updates

.....continued

Section	Description
16-Bit Counter (Mode 1)	Editorial updates
Clock/Calendar (Mode 2)	Updated the section
RTC	Updated the following registers: • SYNCBUSY register in COUNT32 mode • FREQCORR register in COUNT32 mode • COMPn register in COUNT32 mode • GPn register in COUNT32 mode • FREQCORR register in COUNT16 mode • COMPn register in COUNT16 mode • SYNCBUSY register in Clock mode • FREQCORR register in Clock mode • Updated ALARMn register in Clock mode • GPn register in Clock mode
DMAC - Principle of Operation	Removed Burst transfer content
DMAC - Sleep Mode Operation	Updated the section
DMAC	Updated the following register: • CTRL • PENDCH • CHCTRLB • BTCTRL
EIC - Features	Editorial updates
EIC - Interrupts	Added a note
EIC	Updated the following register: • CTRLA • NMICTRL • EVCTRL • INTENCLR • INTENSET • INTFLAG • ASYNCH • CONFIG
NVMCTRL - Features	Updated the number of regions
NVMCTRL - Memory Organization	Added related links
NVMCTRL - NVM User Configuration	Added related links
PORT - Register Description	Added a Tip

.....continued

Section	Description
PORT	Updated the following register: • CTRL • EVCTRL • PMUX • PINCFG
EVSYS - Initialization	Updated the section
EVSYS	Updated the following register: • CHANNELn • USERm
SERCOM USART - Clocks	Added Related Links
SERCOM USART - Initialization	Updated the section
SERCOM SPI	Updated the following register: • INTENCLR • INTENSET
TC	Register Summary and Register Description sections are regrouped
TC	Added the following registers for 16-bit and 32-bit modes: • PER • PERBUF
TC	Updated the following register: • CTRLA • STATUS • CCx for 16-bit and 32-bit mode
TCC	• Updated the following register: – INTENCLR – PER
USB - Overview	Updated the section
USB	Updated the following register: • QOSCTRL • STATUS • EPSTATUSCLRn • EPSTATUSSETn • EPSTATUSn • EPINTFLAGn • STATUS_BK
CCL - Overview	Editorial updates

.....continued

Section	Description
CCL - Signal Description	Updated the pin names and added a note
CCL - Debug Operation	Updated the section
CCL - Principle of Operation	Updated the section
CCL - Initialization	Updated the section
CCL - Internal Events Inputs Selection (EVENT)	Updated the section
ADC- Block Diagram	Removed VREFA signal from the block diagram
ADC - Signal Description	Removed VREFA signal from the *Signal Description* table
ADC - Analog Connections	Updated the section
ADC - Prescaler Selection	Updated the section
ADC - Reference Configuration	Updated the section
ADC - Additional Features	Added new sections
ADC	Updated the REFCTRL register
AC	Updated the following register: • CTRLB • STATUSA • STATUSB • SYNCBUSY • COMPCTRL
PTC - Overview	Updated the section
PTC - Features	Updated the section
RFCTRL - Functional Description	Editorial updates
RFCTRL	Updated the FECTRL register
Reference Guide - AT86RF212B - Spreading, Modulation, and Pulse Shaping	Added Base Band Transmitter Architecture figure
Reference Guide - AT86RF212B - RX/TX Indicator	Updated the section title
Electrical Characteristics - Wake-Up Time	Removed OFF mode details from *Wake-Up Timing* table
Design Considerations	Added a new chapter
Acronyms and Abbreviations	Editorial updates
Document	Rearranged the chapter 6 and 7

24.2 Rev. B - 04/2017

Section	Description
Errata	Added Errata output placement details
SAM R30 Basic Application Schematic	Updated BOM

24.3 Rev. A - 03/2017

Section	Description
Document	Initial Revision

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-7731-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

ATSAMR30G18A-MUT ATSAMR30E18A-MUT ATSAMR30G18A-MU ATSAMR30E18A-MU