UM10876

NHS31xx User manual

Rev. 2.3 — 13 September 2021 User manual

Document information

Information	Content
Keywords	NHS3100, NHS3152, ARM Cortex-M0+, NFC, Temperature, Cold chain, Monitoring, Therapy compliance, Adherence
Abstract	HW user manual for the NHS31xx family of wireless sensor node ICs

Table 1. Revision history

Rev	Date	Description
v2.3	20210913	Updated temperature accuracy numbers.
v2.2	20200910	Updated description of memory map as seen from RF interface.
v2.1	20200420	Added soldering guidelines for HVQFN24 and WLCSP25 ICs.
v2	20180330	Updated issue
Modifications:	• Text and graphics have been updated throughout the document.	
v1	20161130	Initial version

1 Introductory information

1.1 Introduction

The NHS31xx are a family of ICs optimized for monitoring and logging. With their embedded NFC interface, internal temperature sensor and direct battery connection, they support effective system solutions with a minimal number of external components.

The embedded ARM Cortex-M0+ offers flexibility to the users of these ICs to implement their own dedicated solution. The NHS31xx family contains multiple features, like a selectable CPU frequency of up to 8 MHz and various power-down modes for ultra-low-power consumption.

Users can program this NHS31xx family with the industry-wide standard solutions for ARM Cortex-M0+ processors.

Peripheral components include an ultra-low-power RTC, I^2C-bus interface, SPI interface with SSP features, NFC wireless interface and up to 12 general-purpose I/O pins.

Depending on the chip variant, other features include a temperature sensor, specific interfaces for capacitive, resistive and current measurements, and a 12-bit ADC/DAC. There are also specific analog interfaces for interfacing with photodiodes and LEDs.

1.2 Attributes

Note: *Not all features are available in all family members.*

- System:
  - ARM Cortex-M0+ processor, running at frequencies of up to 8 MHz
  - ARM Cortex-M0+ built-in Nested Vectored Interrupt Controller (NVIC)
  - ARM Serial Wire Debug (SWD)
  - System tick timer
  - Fast (single-cycle) multiplier
  - Support for wake-up interrupt controller
  - Vector Table remapping possible
  - IC reset input
- Debug options:
  - Serial Wire Debug with two watchpoint comparators and four breakpoint comparators
  - Halting debug
- Memory:
  - 32 kB on-chip flash programming memory
  - 4 kB on-chip EEPROM of which 256 bytes can be write protected
  - 8 kB SRAM
- Analog peripherals available, depending on variant:
  - Temperature sensor with:
    - ±0.5 °C absolute temperature accuracy between −40 °C and 0 °C
    - ±0.3 °C absolute temperature accuracy between 0 °C and +45 °C
    - ±0.5 °C absolute temperature accuracy between +45 °C and +85 °C
  - Current-to-digital converter
  - SAR ADC
  - DAC for generating external biasing voltages
  - 8-bit 20 mA current DAC on 4 pins

- Flexible analog on-chip switch:
  - Each of the 6 analog I/O pins can be dynamically connected to the on-chip converters.
  - Measuring 6 voltages connected to the 6 analog pins is possible using time-division multiplexing
- Digital peripherals:
  - Up to 12 General-Purpose Input Output (GPIO) pins with configurable pull-up/pull-down resistors
  - GPIO pins that can be used as edge and level sensitive interrupt sources
  - High-current drivers/sinks (20 mA) on four pins
  - High-current drivers/sinks (20 mA) on two I^2C-bus pins
  - Programmable watchdog timer (WDT)
- Communication interfaces:
  - NFC/RFID ISO 14443 type A interface
  - SPI controller with SSP features and with FIFO and multi-protocol capabilities
  - I^2C-bus interface supporting full I^2C-bus specification and Fast-mode with a data rate of 400 kbit/s with multiple address recognition and monitor mode
- Clock generation:
  - 8 MHz internal RC oscillator trimmed to 2 % accuracy that is used as the system clock
  - Timer oscillator operating at 32 kHz linked to On/Off Timer unit, driving the real-time clock timer
- Power control:
  - Supply voltage range: 1.72 V to 3.6 V
  - Passive powering via NFC field possible
  - Integrated Power Management Unit (PMU) for fine-grained control of power consumption
  - Four reduced power modes: Sleep, Deep-sleep, Deep power-down and Battery-off
  - Power gating for each analog peripheral for ultra-low-power operation
  - < 50 nA current consumption with battery power switch open
  - Power-On Reset (POR)
- Unique device serial number for identification
- Wide operating temperature range
- Available in HVQFN24, WLCSP25 packages and as gold plated bumped die.

1.3 Device-dependent features

The following features are available, depending on the chip variant:

Table 2. Feature overview

Variant	Function	ADC / DAC	RTC	NFC	Temp.	I2D	SPI
NHS3100	temperature Logger	-	yes	yes	yes	-	yes
NHS3152	therapy adherence monitor - resistive	yes	yes	yes	yes	yes	yes

1.4 Ordering information

Table 3. Ordering Information

Type number	Package		
	Name	Description	Version
NHS3100	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; 4 × 4 × 0.85 mm	SOT616-3
NHS3100UK	WLCSP25	wafer level chip scale package; 25 balls; 2.51 × 2.51 × 0.5 mm	SOT1401-1
NHS3100W8	bumped die	bumped die with 8 functional bumps; 2.51 × 2.51 × 0.16 mm	SOT1870-1
NHS3152	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; 4 × 4 × 0.85 mm	SOT616-3
NHS3152UK	WLCSP25	wafer level chip scale package; 25 balls; 2.51 × 2.51 × 0.5 mm	SOT1401-1

1.5 Block diagram

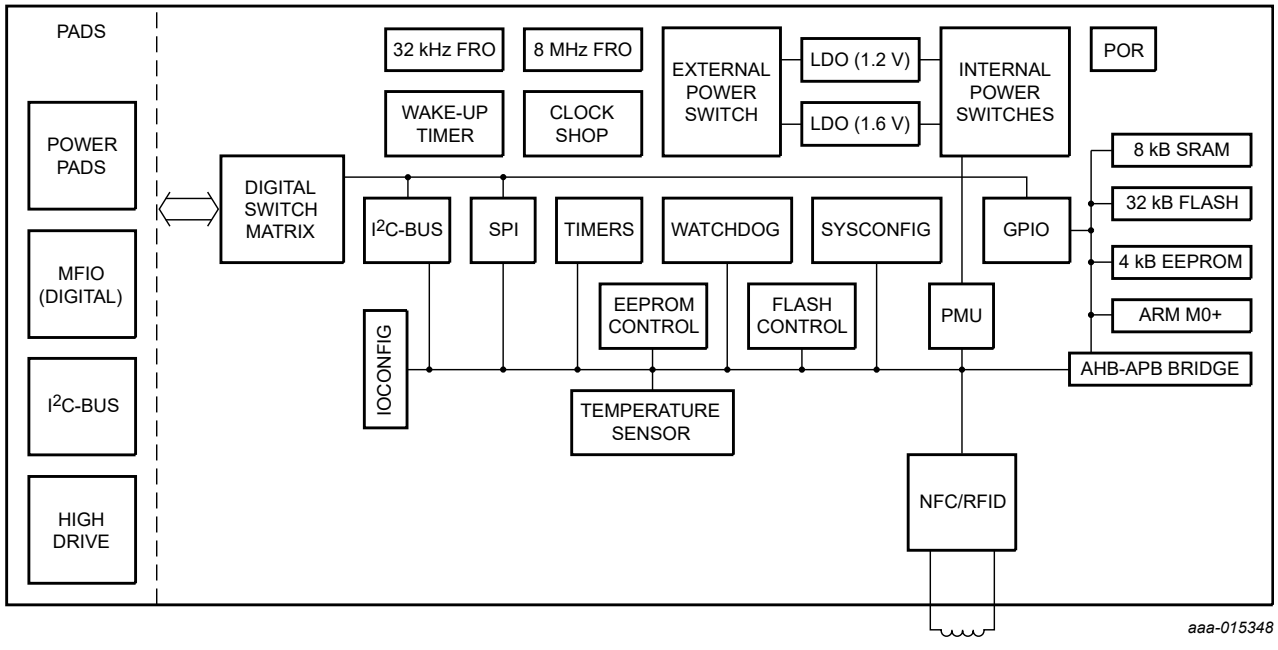


Figure 1. NHS3100 block diagram

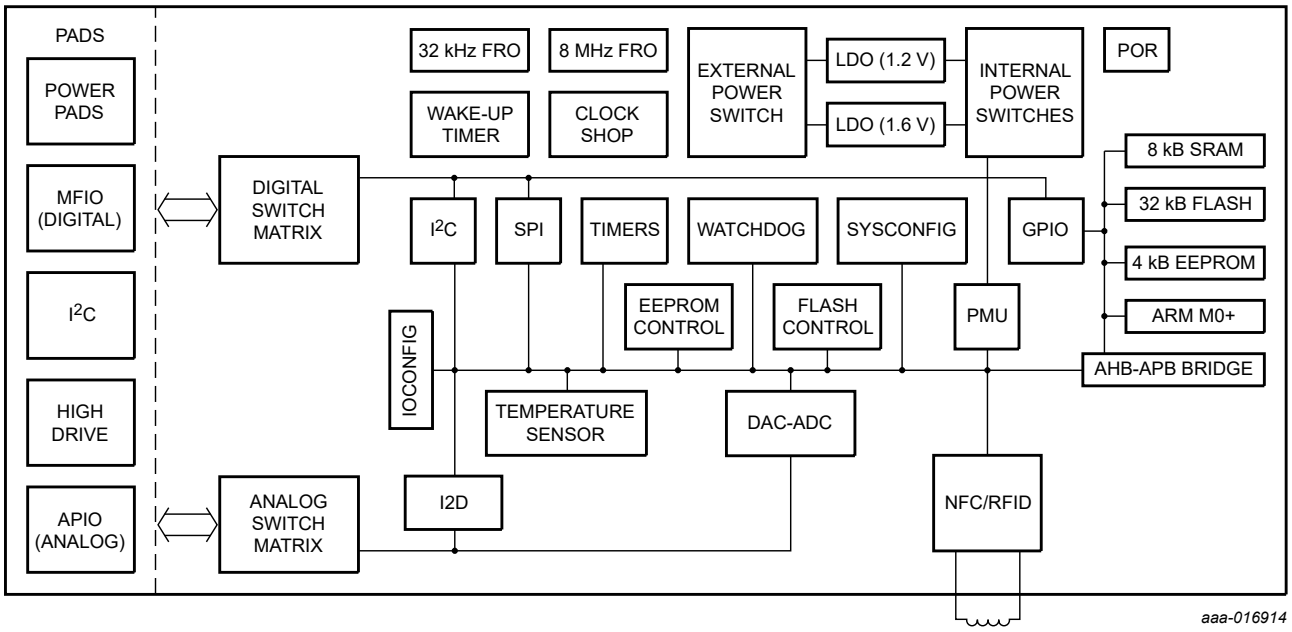


Figure 2. NHS3152 block diagram

1.6 Arm Cortex-M0+ core configuration

The Arm Cortex-M0+ core operates at frequencies up to 8 MHz. Integrated in the core are the NVIC and serial wire debug with four breakpoints and two watch points. It supports a single-cycle I/O enabled port (IOP) for fast GPIO access.

The specific firmware loaded on the non-volatile memory of the device determines its behavior.

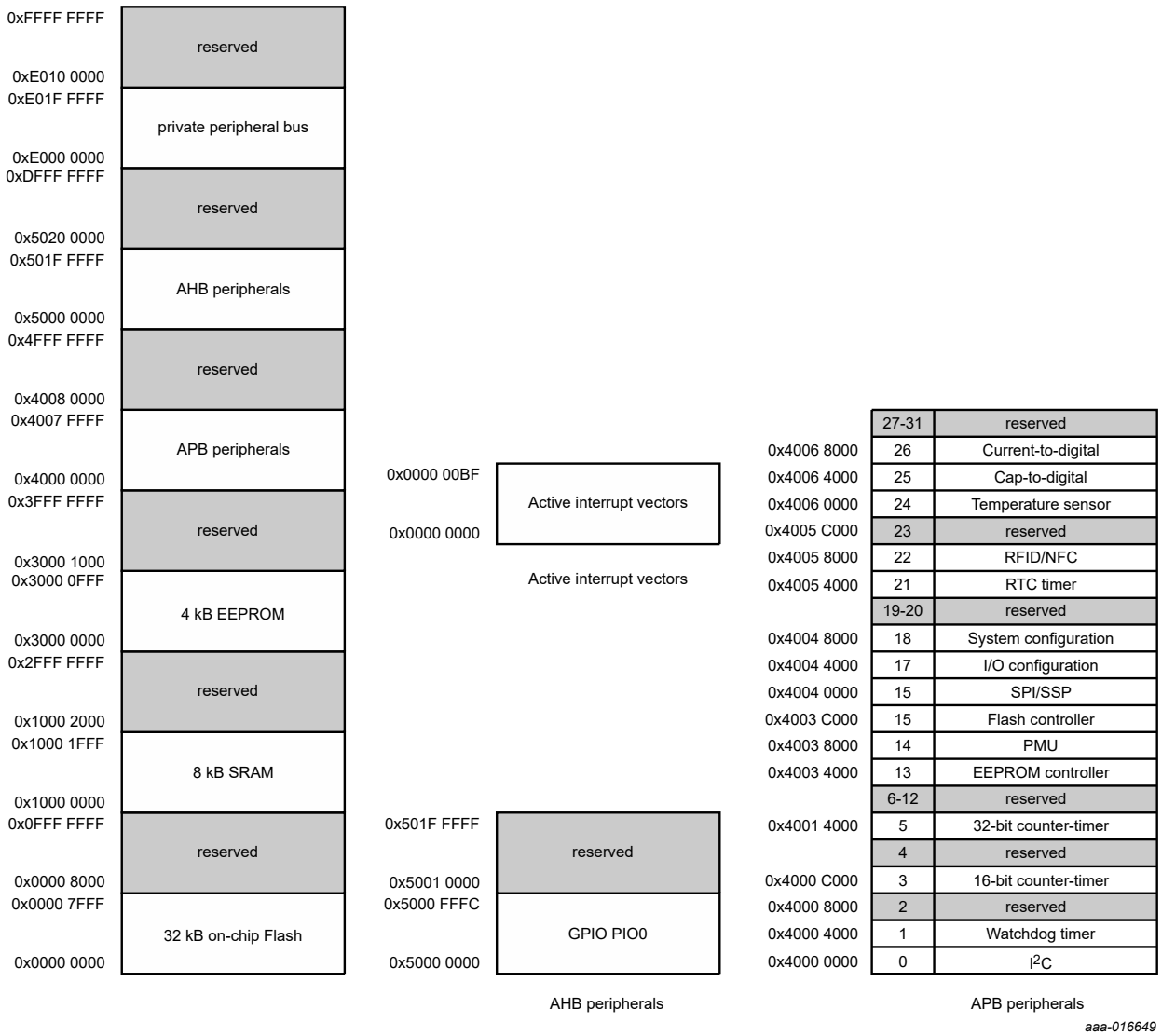
The Arm Cortex-M0+ processor is described in detail in Ref. 3. For the NHS31xx family, the Arm Cortex-M0+ core is configured as follows:

- System options:
  - Nested vectored interrupt controller (NVIC) which is included and supports up to 32 interrupts
  - Fast (single-cycle) multiplier
  - System tick timer
  - Support for wake-up interrupt controller
  - Vector table remapping register
  - Reset all registers present
  - No memory protection unit
  - No single-cycle I/O port
  - Instruction fetch width, mostly 32-bit wide
  - Data endianness: little endian
- Debug options
  - Serial wire debug is included with two watchpoint comparators and four breakpoint comparators
  - Halting debug is supported

1.7 Memory map

Figure 3 shows the memory and peripheral address space.

Peripheral accesses can only be done as 32-bit accesses on addresses divisible by 4. An implication of this is that word and halfword registers must be accessed at the same time. For example, it is not possible to read or write the upper bytes of a word register separately.



Note: Not all registers on the APB block are present on all the chip variants.

Figure 3. NHS31xx family memory map

1.7.1 Memories

1.7.1.1 SRAM (0x1000 0000 to 0x1000 1FFF)

The NHS31xx contains a total of 8 kB on-chip static RAM memory configured as 256 × 2 × 4 × 32 bit. The SRAM supports byte-level access (BWE = 8).

1.7.1.2 Flash (0x0000 0000 to 0x0000 7FFF)

The NHS31xx contains a 32 kB Flash memory of which 30 kB can be used as program and data memory.

The flash is organized in 32 sectors of 1 kB. Each sector consists of 16 rows of 16 × 32-bit words.

1.7.1.3 EEPROM (0x3000 0000 to 0x3000 0FFF)

The NHS31xx contains a 4 kB EEPROM. This EEPROM is organized in 64 rows of 32 × 16-bit words. Of these rows, the last five contain calibration and test data and are locked. This data is either used by the boot loader after reset, or made accessible to the application via firmware API.

2 Product feature overview

2.1 About this section

To suit different applications, the NHS31xx is available in several different variants. This section gives an overview of the principle components of each variant.

2.2 General description

All ICs share flexibility, standard adherence and ultra-low-power features:

- The embedded ARM Cortex-M0+ offers flexibility and processing power to the users of this IC. It enables them to implement their own dedicated solution, including decision making on the IC itself. The ICs contain multiple features including various power-down modes and a selectable CPU frequency up to 8 MHz for ultra-low-power consumption.
- Users can program this IC via SWD using industry-wide standard solutions for ARM Cortex-M0+ processors, or via NFC and the freely provided NFC program downloader. Programming via NFC can only be done once.

2.2.1 NHS3100 for cold chain monitoring and therapy adherence

The NHS3100 is an IC optimized for temperature monitoring and logging. The embedded NFC interface, allows fully NFC-forum standards-compliant communication, internal temperature sensing and direct battery connection. It supports an effective system solution with a minimal of external components.

The NHS3100 works either battery-powered or NFC-powered.

2.2.2 NHS3152 for therapy adherence using resistive sensing

The NHS3152 is an IC optimized for realizing therapy adherence monitoring and logging solutions. The embedded NFC interface fully allows NFC-forum standards-compliant communication, a resistive network sensing interface, an internal temperature sensor and direct battery connection. The NHS3152 enables the making of an effective system solution, supporting a single layer foil implementation, for pill usage monitoring.

The NHS3152 works either battery-powered or NFC-powered.

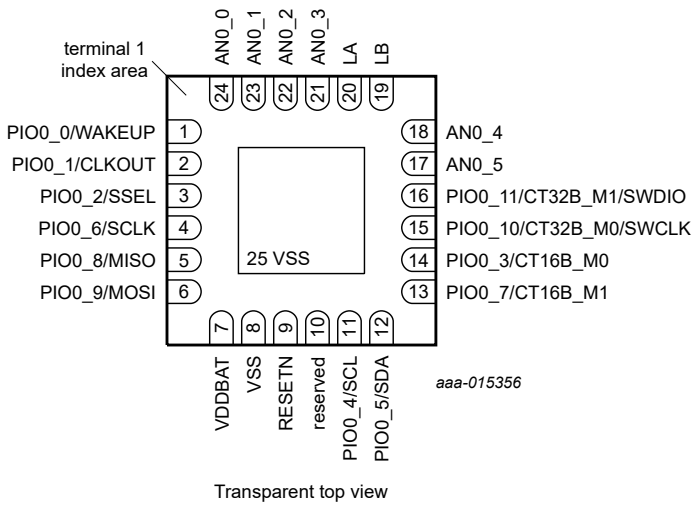
3 Pinning

3.1 About this section

This section describes the pin layout and functions of the NHS31xx family in the HVQFN24 and WLCSP25 packages.

The pin functionality depends on the particular configuration of the chip and is customer or application dependent. Pin functions are software-assigned through the IOCON configuration registers (see Section 23). The pinning of the packages is shown below.

3.2 HVQFN24



Transparent top view

Note: *Not all pins are used in all family variants.*

Figure 4. Pin configuration of HVQFN24 package

Table 4. Pad allocation table of the HVQFN24 package

Pad	Symbol	Pad	Symbol
1	PIO0_0/WAKEUP	13[1]	PIO0_7/CT16B_M1
2	PIO0_1/CLKOUT	14[1]	PIO0_3/CT16B_M0
3	PIO0_2/SSEL	15[1]	PIO0_10/CT32B_M0/SWCLK
4	PIO0_6/SCLK	16[1]	PIO0_11/CT32B_M1/SWDIO
5	PIO0_8/MISO	17[2]	AN0_5
6	PIO0_9/MOSI	18[2]	AN0_4
7	VDDBAT	19	LB
8	VSS	20	LA
9	RESETN	21[2]	AN0_3
10	reserved	22[2]	AN0_2

Table 4. Pad allocation table of the HVQFN24 package...*continued*

Pad	Symbol	Pad	Symbol
11	PIO0_4/SCL	23[2]	AN0_1
12	PIO0_5/SDA	24[2]	AN0_0

[1] High source current pads. See Section 23.2.4.
[2] Only used in variants with analog I/O. In other variants, they must be tied to ground.

Table 5. Pad description of the HVQFN24 package

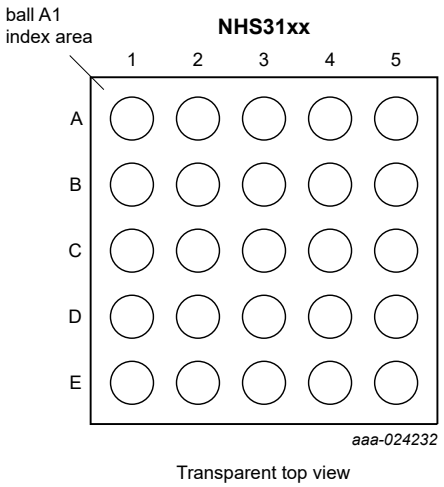
Pad	Symbol	Type	Description
Supply			
7	VDDBAT	supply	positive supply voltage
8	VSS	supply	ground
GPIO[1]			
1	PIO0_0	I/O	GPIO
	WAKEUP	I	deep power-down mode wake-up pin[2]
2	PIO0_1	I/O	GPIO
	CLKOUT	O	clock output
3	PIO0_2	I/O	GPIO
	SSEL	I	SPI/SSP serial select line
14	PIO0_3	I/O	GPIO
	CT16B_M0	O	16-bit timer match output 0
11	PIO0_4	I/O	GPIO[3]
	SCL	I/O	I^2C-bus SCL clock line
12	PIO0_5	I/O	GPIO[3]
	SDA	I/O	I^2C-bus SDA data line
4	PIO0_6	I/O	GPIO
	SCLK	I/O	SPI/SSP serial clock line
13	PIO0_7	I/O	GPIO
	CT16B_M1	O	16-bit timer match output 1
5	PIO0_8	I/O	GPIO
	MISO	O	SPI/SSP master-in slave-out line
6	PIO0_9	I/O	GPIO
	MOSI	I	SPI/SSP master-out slave-in line
15	PIO0_10	I/O	GPIO
	CT32B_M0	O	32-bit timer match output 0
	SWCLK	I	ARM SWD clock

Table 5. Pad description of the HVQFN24 package*...continued*

Pad	Symbol	Type	Description
16	PIO0_11	I/O	GPIO
	CT32B_M1	O	32-bit timer match output 1
	SWDIO	I/O	ARM SWD I/O
Analog I/O[4][5]			
24	AN0_0	A	to AN0_BUS0
23	AN0_1	A	to AN0_BUS1
22	AN0_2	A	to AN0_BUS2
21	AN0_3	A	to AN0_BUS3
18	AN0_4	A	to AN0_BUS4
17	AN0_5	A	to AN0_BUS5
Radio			
20	LA	A	NFC antenna/coil terminal A
19	LB	A	NFC antenna/coil terminal B
Reset			
9	RESETN	I	external reset input[6]

[1] The GPIO port is a 12-bit I/O port with individual direction and function controls for each bit. The operation of port 0 pads depends on the function selected through the IOCONFIG register block.
[2] If external wake-up is enabled on this pad, it must be pulled HIGH before entering Deep power-down mode and pulled LOW for a minimum of 100 µs to exit Deep power-down mode.
[3] Open drain, no pull-up or pull down.
[4] The analog port is a 6-input analog I/O port with enable control for each pad.
[5] Only used in variants with Analog I/O. In other variants, they must be tied to ground.
[6] A LOW on this pad resets the device. This reset causes I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. It has weak pull-up to V_{BAT} or internal NFC voltage (whichever is highest).

3.3 WLCSP25



Transparent top view

Note: *Not all balls are used in all family variants.*

Figure 5. Ball configuration of WLCSP25 package

Table 6. Ball allocation table of the WLCSP25 package

Ball	Symbol	Ball	Symbol
A1	VDDBAT	C4[1]	PIO0_7/CT16B_M1
A2	VSS	C5[1]	PIO0_11/CT32B_M1/SWDIO
A3	RESETN	D1	PIO0_0/WAKEUP
A4	PIO0_4/SCL	D2	PIO0_1/CLKOUT
A5	PIO0_5/SDA	D3[2]	AN0_2
B1	PIO0_8/MISO	D4[2]	AN0_4
B2	PIO0_9/MOSI	D5[2]	AN0_5
B3	reserved	E1[2]	AN0_0
B4[1]	PIO0_3/CT16B_M0	E2[2]	AN0_1
B5[1]	PIO0_10/CT32B_M0/SWCLK	E3[2]	AN0_3
C1	PIO0_2/SSEL	E4	LA
C2	PIO0_6/SCLK	E5	LB
C3	VSS	-	-

[1] High source current balls. See Section 23.2.4.
[2] Only used in variants with Analog I/O. In other variants, they must be tied to ground.

Table 7. Ball description of the WLCSP25 package

Ball	Symbol	Type	Description
Supply			
A1	VDDBAT	supply	positive supply voltage
A2, C3	VSS	supply	ground
GPIO[1]			
D1	PIO0_0	I/O	GPIO
	WAKEUP	I	Deep power-down mode wake-up ball[2]
D2	PIO0_1	I/O	GPIO
	CLKOUT	O	clock output
C1	PIO0_2	I/O	GPIO
	SSEL	I	SPI/SSP serial select line
B4	PIO0_3	I/O	GPIO
	CT16B_M0	O	16-bit timer match output 0
A4	PIO0_4	I/O	GPIO[3]
	SCL	I/O	I^2C SCL clock line
A5	PIO0_5	I/O	GPIO[3]
	SDA	I/O	I^2C-bus SDA data line
C2	PIO0_6	I/O	GPIO
	SCLK	I/O	SPI/SSP serial clock line

Table 7. Ball description of the WLCSP25 package*...continued*

Ball	Symbol	Type	Description
C4	PIO0_7	I/O	GPIO
	CT16B_M1	O	16-bit timer match output 1
B1	PIO0_8	I/O	GPIO
	MISO	O	SPI/SSP master-in slave-out line
B2	PIO0_9	I/O	GPIO
	MOSI	I	SPI/SSP master-out slave-in line
B5	PIO0_10	I/O	GPIO
	CT32B_M0	O	32-bit timer match output 0
	SWCLK	I	ARM SWD clock
C5	PIO0_11	I/O	GPIO
	CT32B_M1	O	32-bit timer match output 1
	SWDIO	I/O	ARM SWD I/O
Analog I/O[4][5]			
E1	AN0_0	A	to AN0_BUS0
E2	AN0_1	A	to AN0_BUS1
D3	AN0_2	A	to AN0_BUS2
E3	AN0_3	A	to AN0_BUS3
D4	AN0_4	A	to AN0_BUS4
D5	AN0_5	A	to AN0_BUS5
Radio			
E4	LA	A	NFC antenna/coil terminal A
E5	LB	A	NFC antenna/coil terminal B
Reset			
A3	RESETN	I	external reset input[6]

[1] The GPIO port is a 12-bit I/O port with individual direction and function controls for each bit. The operation of port 0 pads depends on the function selected through the IOCONFIG register block.
[2] If external wake-up is enabled on this pad, it must be pulled HIGH before entering Deep power-down mode and pulled LOW for a minimum of 100 µs to exit Deep power-down mode.
[3] Open drain, no pull-up or pull down.
[4] The analog port is a 6-input analog I/O port with enable control for each pad.
[5] Only used in variants with Analog I/O. In other variants, they must be tied to ground.
[6] A LOW on this pad resets the device. This reset causes I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. It has weak pull-up to V_{BAT} or internal NFC voltage (whichever is highest).

3.4 NHS3100W8

Figure 6 shows the bump layout of the NHS3100W8 gold bump version.

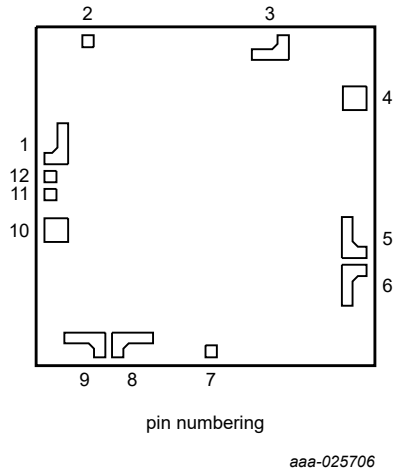


Figure 6. Bump configuration Bump die: Top view, bumps up

Table 8. Bump allocation table of the NHS3100W8 package

Bump	Symbol	Bump	Symbol
1	PIO0_0/WAKEUP	7	TP1
2	TP0	8	VSS
3	LA	9	VDDBAT
4	LB	10	PIO0_6
5	PIO0_11/CT32B_M1/SWDIO	11	TP2
6	PIO0_10/CT32B_M0/SWCLK	12	TP3

Table 9. Bump description of the NHS3100W8 package

Bump	Symbol	Type	Description
Supply			
9	VDDBAT	supply	positive supply voltage
8	VSS	supply	ground
GPIO[1]			
1	PIO0_0	I/O	GPIO
	WAKEUP	I	Deep power-down mode wake-up pin[2]
10	PIO0_6	I/O	GPIO
6	PIO0_10	I/O	GPIO
	CT32B_M0	O	32-bit timer match output 0
	SWCLK	I	ARM SWD clock

Table 9. Bump description of the NHS3100W8 package*...continued*

Bump	Symbol	Type	Description
5	PIO0_11	I/O	GPIO
	CT32B_M1	O	32-bit timer match output 1
	SWDIO	I/O	ARM SWD I/O
Radio			
3	LA	A	NFC antenna/coil terminal A
4	LB	A	NFC antenna/coil terminal B
Test pins			
2	TP0	-	test pin - do not connect
7	TP1	-	test pin - do not connect, or connect to ground
11	TP2	-	test pin - do not connect
12	TP3	-	test pin - do not connect

[1] The GPIO port is a 12-bit I/O port with individual direction and function controls for each bit. The operation of port 0 balls depends on the function selected through the IOCONFIG register block.

[2] If external wake-up is enabled on this ball, it must be pulled HIGH before entering Deep power-down mode and pulled LOW for a minimum of 100 µs to exit Deep power-down mode.

4 System configuration (SYSCON)

4.1 General description

The system configuration block is at APB 0x4004 8000. It controls oscillators, start logic, and clock generation of the NHS31xx. It contains registers controlling power on/off of the peripherals and a register to remap the ARM vector table to a selectable location in Flash or SRAM.

4.2 Clock generation

The NHS31xx clock generator unit (CGU) includes two independent RC oscillators. These oscillators are the System Free-Running Oscillator (SFRO) and the Timer Free-Running Oscillator (TFRO).

The SFRO is a current-controlled oscillator and runs at 8 MHz. The system clock is derived from it. The system clock can be set to 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, or 62.5 kHz with the SYSCLKCTRL system configuration register (see Table 13). The default system clock operating frequency is 500 kHz to minimize dynamic current consumption during the boot cycle.

Note: *When the lower clock speeds are used, some features are not available.*

The TFRO runs at 32.768 kHz and is the clock source for the Real-Time Clock (RTC) unit and the power management unit. The TMRCLKCTRL configuration register similarly controls it.

Following reset, the NHS31xx starts operating from the SFRO at the default 500 kHz clock frequency. The user can change the system clock frequency by setting the SYSCLKSEL. The SFRO cannot be disabled.

The SYSAHBCLKCTRL register gates the system clock to the various peripherals and memories. The SPI unit has an individual clock divider to derive the serial clock from the SFRO. The watchdog timer unit also has an individual clock divider.

The analog parts of the analog-to-digital converters receive a fixed clock frequency, irrespective of the system clock divider settings. The digital part uses the APB clock. The analog part of the temperature sensor receives 2 MHz, the ADC/DAC receives 8 MHz, and the other peripherals 1 MHz.

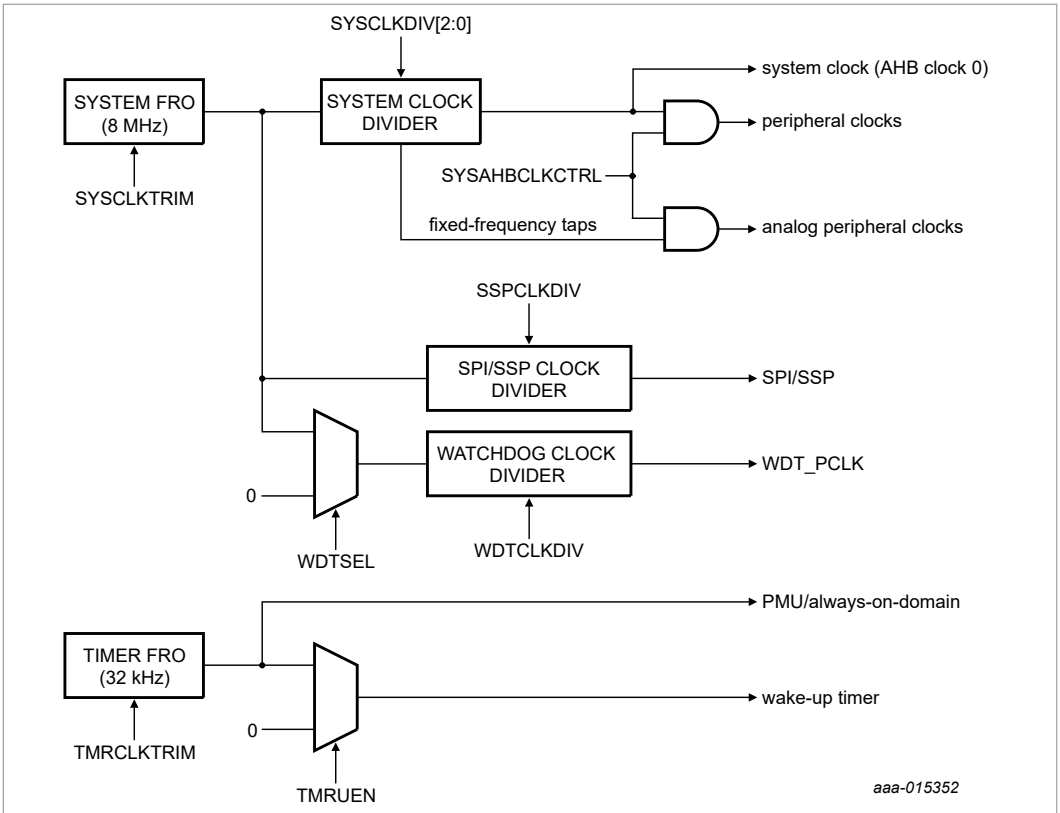


Figure 7. NHS31xx clock generator block diagram

4.3 System FRO clock specifications

The SFRO provides the main system clock for the NHS31xx. The system clock is enabled by default and cannot be disabled.

4.3.1 SFRO trimming

The 8 MHz SFRO is trimmed by setting the SYSCLKTRIM bits in the SYSCLKCTRL register. Trimming changes the internal biasing current regulating the oscillation frequency. 1 LSB of SYSCLKTRIM corresponds to approximately ±0.65 % change in oscillation frequency.

4.4 Timer FRO clock specifications

The TFRO provides a 32.768 kHz signal to the Real Time Clock (RTC) unit.

4.5 Register descriptions

All registers, regardless of size, are on addresses divisible by 4. Details of the registers appear in the description of each function.

See Section 18.5 for the flash access timing register. This register can be reconfigured as part of the system setup. Also see Section 17.4 for the EEPROM.

Table 10. Register overview system configuration control block - SYSCON (base address 0x4004 8000)

Name	Access	Address offset	Description	Reset values	Reference
SYSMEMREMAP	R/W	0x000	system memory remap	0x0000 0000	Table 11
PRESETCTRL	R/W	0x004	peripheral reset control	0x0000 0000	Table 12
-	-	0x008 - 0x01C	reserved	-	-
SYSCLKCTRL	R/W	0x020	system clock control register	0x0002 0008	Table 13
SYSCLKUEN	R/W	0x024	system clock update enable	0x0000 0000	Table 15
-	-	0x028 - 0x02C	reserved	-	-
SYSRSTSTAT	R/W	0x030	system reset status register	0x0000 0000	Table 16
-	-	0x034 - 0x078	reserved	-	-
SYSAHBCLKCTRL	R/W	0x080	AHB clock control	0x0001 C007	Table 17
-	-	0x084 - 0x090	reserved	-	-
SSPCLKDIV	R/W	0x094	SPI/SSP clock divider	0x0000 0000	Table 18
-	-	0x098 - 0x0CC	reserved	-	-
WDTCLKSEL	R/W	0x0D0	watchdog timer clock selector	0x0000 0000	Table 19
WDTCLKUEN	R/W	0x0D4	watchdog timer clock update enable	0x0000 0000	Table 20
WDTCLKDIV	R/W	0x0D8	watchdog timer clock divider	0x0000 0000	Table 21
-	-	0x0DC - 0x0E4	reserved	-	-
CLKOUTEN	R/W	0x0E8	CLKOUT enable	0x0000 0000	Table 22
-	-	0x0EC - 0x150	reserved	-	-
SYSTCKCAL	R/W	0x154	system tick counter calibration	0x0000 0004	Table 23
-	-	0x210 - 0X230	reserved	-	-
STARTAPRP0	R/W	0x200	start logic edge control register 0	0x0000 0000	Table 24
STARTERP0	R/W	0x204	start logic signal enable register 0	0x0000 0000	Table 25
STARTRSRP0CLR	R/W	0x204	start logic reset register 0	0x0000 0000	Table 26
STARTSRP0	R	0x20C	start logic status register 0	0x0000 0000	Table 27
-	-	0x210 - 0X230	reserved	-	-
PDAWAKECFG	R/W	0x234	Flash address mapping after wake-up from Deep-sleep mode	0x0000 0000	Table 28
PDRUNCFG	R/W	0x238	power-down configuration register	0x0000 003E	Table 29
-	-	0x23C - 0x3EC	reserved	-	-

Table 10. Register overview system configuration control block - SYSCON (base address 0x4004 8000)*...continued*

Name	Access	Address offset	Description	Reset values	Reference
DEVICE_ID	R	0x3F4	-	-	Table 30

4.5.1 System memory remap register (SYSMEMREMAP)

The system memory remap register contains the addresses of the ARM interrupt vectors. Valid addresses are on 1024-byte boundaries in Flash or SRAM. The first 192 addresses (0 to 191) are remapped to the locations indicated below.

Table 11. SYSMEMREMAP register (address 0x4004 8000) bit description

Bit	Symbol	Description		After boot
0	map	interrupt vector remap		0
		0	interrupt vectors reside in Flash	
		1	interrupt vectors reside in SRAM	
5:1	offset	system memory remap offset		00000b
		00000b	interrupt vectors in flash or remapped to SRAM but not offset	
		00001b - 00111b	interrupt vectors offset in flash or SRAM to 1 K word segment	
		01000b - 11111b	interrupt vectors offset in flash to 1 K word segment 8 to 31	
31:6	-	reserved		0

4.5.2 Peripheral reset control register (PRESETCTRL)

This register allows software to reset some peripherals. Writing a logic 0 to the bits resets the corresponding peripheral. Writing a logic 1 de-asserts the reset.

Note: *Before accessing the SPI and I^2C peripherals, write a logic 1 to this register. This action ensures that the reset signals to the SPI and I^2C are de-asserted.*

Table 12. PRESETCTRL register (address 0x4004 8004) bit description

Bit	Symbol	Description		After boot
0	SSP_RST_N	SPI/SSP reset control		0
		0	resets the SPI peripheral	
		1	SPI reset de-asserted	
1	I2C_RST_N	I^2C-bus reset control		0
		0	resets the I^2C-bus peripheral	
		1	I^2C-bus reset de-asserted	
2	EE_RST_N	EEPROM NVMC reset control		1
		0	resets the EEPROM NVMC controller	
		1	reset de-asserted	

Table 12. PRESETCTRL register (address 0x4004 8004) bit description...*continued*

Bit	Symbol	Description		After boot
3	NFC_RST_N	NFC shared memory reset control		1
		0	resets the NFC shared memory	
		1	reset de-asserted	
31:4	-	reserved		0

4.5.3 System clock control register (SYSCLKCTRL)

The system clock control register controls the analog part of the internal 8 MHz oscillator as well as the system clock divider.

For changes to the system clock divider to take effect, first write a logic 0 to the SYSCLKUEN register, then write a logic 1 to SYSCLKUEN.

Table 13. SYSCLKCTRL register (address 0x4004 8020) bit description

Bit	Symbol	Description		After boot
0	-	reserved. Do not write to this bit.		0
3:1	SYSCLKDIV	system clock divider		100b
		000b	no division (system clock set to 8 MHz)	
		001b	divide-by-2 (system clock set to 4 MHz)	
		010b	divide-by-4 (system clock set to 2 MHz)	
		011b	divide-by-8 (system clock set to 1 MHz)	
		100b	divide-by-16 (system clock set to 500 kHz)	
		101b	divide-by-32 (system clock set to 250 kHz)	
		110b	divide-by-64 (system clock set to 125 kHz)	
		111b	divide-by-128 (system clock set to 62.5 kHz)	
15:4	-	reserved		-
21:16	SYSCLKTRIM	SFRO trim value (6-bit)		device dependent
31:22	-	reserved		-

Table 14 shows the impact on the system of the clock frequency choices.

Table 14. Impact of different clock frequency settings

Setting	Frequency	Impact on system behavior	
000b	8 MHz	I^2C-bus	Standard and Fast mode possible
		flash	LPM requires 1 additional wait state, no wait state needed in High-power mode
		EEPROM	EECLKDIV = 21
001b	4 MHz	I^2C-bus	Standard mode only
		flash	no wait-states needed
		EEPROM	EECLKDIV = 10
010b	2 MHz	I^2C-bus	Standard mode only
		flash	no wait-states needed
		EEPROM	EECLKDIV = 5
011b	1 MHz	I^2C-bus	Standard mode only
		flash	no wait-states needed
		EEPROM	EECLKDIV = 2
100b	500 kHz	I^2C-bus	not supported
		flash	no wait-states needed
		EEPROM	EECLKDIV = 1
101b	250 kHz	I^2C-bus	not supported
		flash	no wait-states needed
		EEPROM	program/erase not supported
110b	125 kHz	I^2C-bus	not supported
		flash	no wait-states needed
		EEPROM	program/erase not supported
111b	62.5 kHz	I^2C-bus	not supported
		flash	no wait-states needed
		EEPROM	program/erase not supported

4.5.4 System clock update enable register (SYSCLKUEN)

This register updates the clock source of the system with the new input clock after the SYSCLKCTRL register has been written to. For the update to take effect, first write a logic 0 to the SYSCLKUEN register and then write a logic 1 to SYSCLKUEN.

Table 15. SYSCLKUEN register (address 0x4004 8024) bit description

Bit	Symbol	Description		Reset value
0	ENA	enable system clock source update		0
		0	no update	
		1	update clock source and clock divisor	
31:1	-	reserved		0

4.5.5 System reset status register (SYSRSTSTAT)

The SYSRSTSTAT register shows the source of the latest reset event. The bits are cleared by writing a value to the register. The POR event clears all other bits in this register. However, if another reset signal - for example EXTRST - remains asserted after the POR signal is negated, then its bit is set to detected.

Table 16. SYSRSTSTAT register (address 0x4004 8030) bit description

Bit	Symbol	Description		Reset value
0	POR [1]	POR reset status		0
		0	no POR detected	
		1	POR detected	
1	EXTRST [1]	status of external RESETN pin		0
		0	no RESETN event detected	
		1	external reset detected	
2	WDT [2]	status of the Watchdog reset		0
		0	no WDT reset detected	
		1	WDT reset detected	
3	SYSRST [3]	status of software system reset (ARM AIRCR register SYSRESETREQ bit was set)		0
		0	no software reset detected	
		1	software reset detected	
31:4	-	reserved		-

[1] POR and EXTRST trigger full system reset.
[2] If the debugger is not active, then WDT triggers full system reset.
[3] SYSRST triggers a reset of the ARM core (the peripheral registers are not reset).

4.5.6 AHB clock control register

The SYSAHBCLKCTRL register enables the clocks to individual system and peripheral blocks. The system clock provides the clock for the AHB to APB bridge, the AHB matrix, the ARM Cortex-M0+, the SYSCON block, and the PMU. This clock cannot be disabled.

Table 17. SYSAHBCLKCTRL register (address 0x4004 8080) bit description

Bit	Symbol	Description		Reset value
0	-	reserved		1
1	-	reserved		-
2	RAM	enable clock for RAM		1
		0	disable	
		1	enable	
4:3	FLASHREG / FLASHARRAY	enable clock for Flash register interface and array access		0b11
		0b00	disable	
		0b01	enable	
		0b10	enable	
		0b11	enable	
5	I2C	enable clock for I^2C-bus		0
		0	disable	
		1	enable	
6	GPIO	enable clock for GPIO		0
		0	disable	
		1	enable	
7	SPISSP	enable clock for SPI/SSP		0
		0	disable	
		1	enable	
8	CT16B	enable clock for 16-bit timer		0
		0	disable	
		1	enable	
9	CT32B	enable clock for 32-bit timer		0
		0	disable	
		1	enable	
10	RTC	enable clock for RTC (APB side only, TFRO directly clocks the timer unit, as set by TMRCLKCTRL register).		0
		0	disable	
		1	enable	
11	WDT	enable clock for watchdog timer		0
		0	disable	
		1	enable	

Table 17. SYSAHBCLKCTRL register (address 0x4004 8080) bit description *...continued*

Bit	Symbol	Description		Reset value
12	TSEN	enable clock for temperature sensor		0
		0	disable	
		1	enable	
13	-	reserved		
14	-	reserved		
15	I2D	enable clock for Current-to-Digital converter		0
		0	disable	
		1	enable	
16	ADC/DAC	enable clock for ADC/DAC converter		0
		0	disable	
		1	enable	
17	-	reserved		
18	IOCON	enable clock for I/O configuration block		1
		0	disable	
		1	enable	
20:19	EEREG / EEARRAY	enable clock for EEPROM register interface and array access		0b11
		0b00	disable	
		0b01	enable	
		0b10	enable	
		0b11	enable	
31:21	-	reserved		0

4.5.7 SPI/SSP clock divider control register (SSPCLKDIV)

This register configures the SPI peripheral clock SPI_PCLK. The SPI_PCLK can be shut down by setting the DIV bits to logic 0.

Table 18. SSPCLKDIV register (address 0x4004 8094) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV	-	SPI_CLK clock divider values: (0) disable, (1) equal to SFRO frequency (even 2-254) divide-by-DIV	0
31:8	-	-	reserved	0

4.5.8 Watchdog timer clock source selection register (WDTCLKSEL)

This register selects the clock source for the watchdog timer. The WDTCLKUEN register must be toggled from logic 0 to logic 1 for the update to take effect.

Table 19. WDTCLKSEL register (address 0x4004 80D0) bit description

Bit	Symbol	Description		Reset value
1:0	SEL	WDT clock source		0
		0x0	system FRO	
		0x1	reserved	
		0x2	disabled	
		0x3	reserved	
31:2	-	reserved		0

4.5.9 Watchdog timer clock update enable register (WDTCLKUEN)

This register updates the clock source of the watchdog timer with the new input source after the WDTCLKSEL register has been written to. For the update to take effect at the input of the watchdog timer, the following procedure must be applied:

1. Write a logic 0 to the WDTCLKUEN register
2. Write a logic 1 to WDTCLKUEN to preset the value
3. Write a logic 0 again to apply the value

The 0-1-0 pulse length should be longer than the period of the actual and selected clock sources.

Table 20. WDTCLKUEN register (address 0x4004 80D4) bit description

Bit	Symbol	Description		Reset value
0	ENA	enable WDT clock source update		0
		0	no update	
		1	update clock source	
31:1	-	reserved		0

4.5.10 Watchdog timer clock divider register (WDTCLKDIV)

This register determines the divider values for the watchdog clock wdt_clk. For the update to take effect at the input of the watchdog timer, a 1-0-1 sequence has to be written to register WDTCLKEN (see Section 4.5.9).

When the division factor is set to 0, the WDT_CLK is disabled. Apart from 1, only even division factors can be used.

Table 21. WDTCLKDIV register (address 0x4004 80D8) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV	1	WDT clock divider values: (0) disable WDT_PCLK, (1) equal to SFRO frequency (even 2-254) divide-by-DIV	0
31:8	-	-	reserved	0

4.5.11 Clock output enable register (CLKOUTEN)

This register enables output of the different clocks to the CLKOUT pin.

Table 22. CLKOUTEN register (address 0x4004 80E8) bit description

Bit	Symbol	Value	Description	Reset value
0	CLKOUTEN	0	disable output	0
		1	enable output	
2:1	CLKOUTSRC	00b	output SFRO (8 MHz)	0
		01b	output system clock (62.5 kHz to 8ηMHz)	
		10b	output TFRO (32 kHz)	
		11b	output NFC clock (1.695 MHz)[1]	
31:3	-	-	reserved	0

[1] The NFC clock is only available if a 13.56 MHz signal is seen at the LA/LB inputs.

4.5.12 System tick counter calibration register (SYSTCKCAL)

This register determines the value of the SYST_CALIB register.

Table 23. SYSTCKCAL register (address 0x4004 8154) bit description

Bit	Symbol	Value	Description	Reset value
25:0	CAL	1	system tick timer calibration value	26'h0004
31:8	-	-	reserved	0

4.5.13 Start logic edge control register 0 (STARTAPRP0)

The STARTAPRP0 register controls the start logic inputs of the digital PIO ports (PIO0_0 to PIO0_10). This register selects a falling or rising edge on the corresponding PIO0 input trigger the start logic.

Every bit in the STARTAPRP0 register controls one port input and is connected to one wake-up interrupt in the NVIC. Bit 0 in the STARTAPRP0 register corresponds to interrupt 0, bit 1 to interrupt 1, etc. (see Table 274), up to a total of 11 interrupts.

Table 24. STARTAPRP0 register (address 0x4004 8200) bit description

Bit	Symbol	Description		Reset value
10:0	APRPIO_n	edge select for start logic input PIO0_n (PIO0_10 to PIO0_0)		0
		0	falling edge (per bit)	
		1	rising edge	
31:11	-	reserved		0

4.5.14 Start logic signal enable register 0 (STARTERP0)

The STARTERP0 register enables or disables the start signal bits in the start logic. The bit assignment for bits 10:0 is identical to Table 24. Bit 11 enables start on RFID, bit 12 enables start on wake-up timer.

Table 25. STARTERP0 register (address 0x4004 8204) bit description

Bit	Symbol	Description		Reset value
10:0	ERPIO_n	enable start signal for start logic input PIO0_n (PIO0_10 to PIO0_0)		0
		0	disabled	
		1	enabled	
11	ERRFID	enable start signal for start logic input RFID		0
		0	disabled	
		1	enabled	
12	ERTMR	enable start signal for start logic input RTC timer		0
		0	disabled	
		1	enabled	
31:13	-	reserved		0

4.5.15 Start logic reset register 0 (STARTRSRP0CLR)

Writing a one to a bit in the STARTRSRP0CLR register resets the start logic state. The bit assignment is identical to Table 25. The start-up logic uses the input signals to generate a clock edge for registering a start signal. This clock edge (falling or rising) sets the interrupt for waking up from Deep-sleep mode. Writing a value to the registers clears the conditions. After writing a logic 1 to the register, write a logic 0 to it to clear the list again.

Table 26. STARTRSRP0CLR register (address 0x4004 8208) bit description

Bit	Symbol	Description		Reset value
10:0	RSRPIO_n	start signal reset for start logic input PIO0_x (PIO0_10 to PIO0_0)		0
		0	do nothing	
		1	writing logic 1 resets the start signal	
11	RSRRFID	start signal reset for start Logic input RFID		0
		0	do nothing	
		1	writing logic 1 resets the start signal	
12	RSRTMR	start signal reset for start logic input wake-up timer		0
		0	do nothing	
		1	writing logic 1 resets the start signal	
31:13	-	reserved		0

4.5.16 Start logic status register 0 (STARTSRP0)

This register reflects the status of the enabled start signal bits. Each bit (if enabled) reflects the state of the start logic, i.e. whether a wake-up signal has been received for a given pin or not.

Table 27. STARTSRP0 register (address 0x4004 820C) bit description

Bit	Symbol	Description		Reset value
10:0	SRPIO_n	start signal status for start logic input PIO0_n (PIO0_10 to PIO0_0)		0
		0	no start signal received	
		1	start signal received	
11	SRRFID	start signal status for start logic input RFID		0
		0	no start signal received	
		1	start signal received	
12	SRTMR	start signal status for start logic input wake-up timer		0
		0	no start signal received	
		1	start signal received	
31:13	-	reserved		0

4.5.17 Wake-up configuration register (PDAWAKECFG)

The FLASH_PD bit in this register, PDAWAKECFG, enables an address remap of the flash addresses to SRAM addresses after exiting from Deep-sleep mode.

When the bit is set before going in Deep-sleep mode, all the flash addresses are remapped to SRAM addresses, when the chip leaves Deep-sleep mode. All the flash addresses are remapped, not only the interrupt vector table. The remapping is independent of the FLASH_PD bit in the PDRUNCFG register. In other words, the remapping is active whether the flash is powered down or not.

The remapping becomes only active when the chip goes in Deep-sleep mode. The remapping continues until the FLASH_PD bit in this register is cleared.

If the user chooses to enable this feature, the interrupt vector table and the program code must be copied to SRAM. The SRAM content is preserved during Deep-sleep mode.

Table 28. PDAWAKECFG register (address 0x4004 8234) bit description

Bit	Symbol	Description		Reset value
0	FLASH_PD	Flash addresses remapped to SRAM after Deep-sleep mode		0
		0	no flash address remapping after Deep-sleep mode (Flash must be powered)	
		1	Flash address remapping after Deep-sleep mode (Flash can be powered down)	
31:1	-	reserved		0

4.5.18 Power-down configuration register (PDRUNCFG)

The bits in the PDRUNCFG register control the power to the various analog blocks. This register can be written to at any time while the chip is powered, and a write takes effect immediately. By default, the Flash memory is running. Before switching off power to Flash memory, the user must ensure the interrupt vector table points to SRAM, and the relevant program code resides in SRAM.

Table 29. PDRUNCFG register (address 0x4004 8238) bit description

Bit	Symbol	Description		Reset value
0	FLASH_PD	flash power down		0
		0	powered	
		1	powered down	
1	TSEN_PD	temperature sensor power down		1
		0	powered	
		1	powered down	
2	-	reserved		
3	EEPROM_PD	EEPROM power down		1
		0	powered	
		1	powered down	
4	I2D	current-to-digital converter power down		1
		0	powered	
		1	powered down	
5	ADCDAC	ADC/DAC converter power down		1
		0	powered	
		1	powered down	
31:6	-	reserved		0

4.5.19 Device ID register (DEVICE_ID)

This device ID register is a read-only register and contains the part ID of the chip (NHS3100, NHS3152...). The value of this register is copied from EEPROM as part of device boot (firmware).

Table 30. DEVICE_ID register (address 0x4004 83F4) bit description

Bit	Symbol	Value	Description	After boot
31:0	DEVICEID	-	part ID number for NHS31xx model	device dependent

Table 31. Device IDs

Type number	Device ID
NHS3100/A1	0x4E310020
NHS3100UK/A1	
NHS3100W8/A1	
NHS3152/A1	0x4E315220
NHS3152UK/A1	

5 Power management unit (PMU)

5.1 About this section

This section describes the power management unit (PMU) which controls the switching between available power sources.

5.2 General description

The PMU controls four power modes:

- Active
- Sleep
- Deep-sleep
- Deep power-down

Additionally, the chip can be put into the Battery-off mode, which reduces current to nA levels. However, this mode can only be left via the RESETN pin or an NFC signal.

The IC power controller controls power flow from the NFC domain or the external domain to the internal domains; the PMU controls the power regulators.

5.3 Functional description

The NHS31xx supports various power control features. In Active mode, when the chip is running, power and clocks to selected peripherals can be optimized for power consumption. In addition, there are three special modes of processor power reduction: Sleep mode, Deep-sleep mode, and Deep power-down mode.

Note: *Basic RFID/NFC tag functionality is maintained in all sleep and power-down modes, as the RFID/NFC communication block is self-powered. Communication with the Arm Cortex-M0+ core is however only possible in Active mode.*

Note: *The Debug mode is not supported in Deep power-down mode.*

5.3.1 Active mode

Active mode is the only mode in which the Arm core is executing instructions. The system clock, or a dedicated peripheral clock, clocks the peripherals. The chip is in Active mode after reset. The reset values of the PDRUNCFG and SYSAHBCLKCTRL registers determine the default power configuration (see Section 4.5.6). The power configuration can be changed during runtime.

5.3.1.1 Power configuration in Active mode

The following configuration choices determine the power consumption in Active mode:

- The SYSAHBCLKCTRL register controls which memories and peripherals are running
- The power to various analog blocks (sensors, ADC, DAC, and the flash block) can be controlled at any time individually through the PDRUNCFG register
- The clock source for the system clock can be selected from the SFRO
- The SYSCLKCTRL register selects the system clock frequency
- Selected peripherals (SPI, WDT) use individual peripheral clocks with their own clock dividers. The peripheral clocks can be shut down through the corresponding clock divider registers

5.3.2 Sleep mode

In Sleep mode, the system clock to the Arm Cortex-M0+ core is stopped. Execution of instructions is suspended until either a reset or an enabled interrupt occurs. Peripheral functions, if selected to be clocked in the SYSAHBCLKCTRL register, continue operation during Sleep mode. These functions may generate interrupts causing the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and their related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static.

The same settings as in Active mode configure the power consumption in Sleep mode:

- Clock remains running
- System clock frequency is the same, but core is not clocked
- Peripherals are controlled as in Active mode

The following steps must be performed to enter Sleep mode:

1. The DPDEN bit in the PCON register must be set to logic 0
2. The SLEEPDEEP bit in the Arm Cortex-M0+ SCR register must be set to logic 0
3. Use the Arm Cortex-M0+ Wait-For-Interrupt (WFI) instruction

Sleep mode is exited automatically when an interrupt enabled by the NVIC arrives at the processor or a reset occurs. After wake-up due to an interrupt, the microcontroller returns to its original power configuration defined by the contents of the PDRUNCFG and the SYSAHBCLKDIV registers. If a reset occurs, the microcontroller enters the default configuration in Active mode.

5.3.3 Deep-sleep mode

In Deep-sleep mode, the system clock to the processor is disabled as in Sleep mode. All analog blocks except for the LDOs and the SFRO are powered down.

Deep-sleep mode eliminates all power used by the flash and analog peripherals, dynamic power used by the processor, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static.

The SFRO keeps running in Deep-sleep mode. The watchdog timer or one of the timers can be left running in Deep-sleep mode if necessary for timer-controlled wake-up.

The following steps must be performed to enter Deep-sleep mode:

1. The DPDEN bit in the PCON register must be set to logic 0
2. If an external pin is used for wake-up, enable and clear the wake-up pin in the start logic registers, and enable the start logic interrupt in the NVIC
3. In the SYSAHBCLKCTRL register, disable all peripherals except timer or WDT if needed
4. Write one to the SLEEPDEEP bit in the Arm Cortex-M0+ SCR register
5. Use the Arm WFI instruction

The microcontroller can wake up from Deep-sleep mode in one of the following ways:

- Signal on an external pin. For this purpose, pins PIO0_0 to PIO0_10 can be enabled as inputs to the start logic. The start logic does not require any clocks
- Input signal to the start logic created by a match event on one of the general-purpose timer external match outputs. The pin holding the timer match function must be enabled as a start logic input in the NVIC. The corresponding timer must be enabled in the SYSAHBCLKCTRL register
- Reset or interrupt from the watchdog timer. In this case, the WDT must be enabled in the SYSAHBCLKCTRL register
- A reset signal from the external RESET pin

The Deep-sleep mode is exit when the start logic indicates an interrupt to the Arm core. The port pins PIO0_0 to PIO0_10, the RFID core, and RTC Timer are connected to the start logic and serve as wake-up sources. The user must program the start logic registers for each input to set the appropriate edge polarity for the corresponding wake-up event. Furthermore, the interrupts corresponding to each input must be enabled in the NVIC.

The start logic does not require a clock to run because it uses the input signals on the enabled pins to generate a clock edge, when enabled. Therefore, the start logic register should be cleared before use.

The start logic can also be used in Active mode to provide a vectored interrupt using the NHS31xx input pins.

5.3.4 Deep power-down mode

In Deep power-down mode, power and clocks are shut off on the internal power domains except for the WAKEUP pin, and the RTC Timer. During Deep power-down mode, the contents of the SRAM and registers are not retained. An exception is a small amount of data which can be stored in the five 32-bit general-purpose registers of the PMU block. SFRO and the two LDOs are shut down, only the always-on domain remains powered (and the NFC/RFID domain, which is self-powered).

The state of the optional pull-up/pull-down on the PIO pins is retained. It can be used to prevent the pins from floating and causing extra power consumption. The I^2C pins (PIO0_4/PIO0_5) do not cause extra power consumption while floating.

Note: *The PIO pins go to 3-state condition WITHOUT pull-up or pull-down when the chip is reset (both from internal or external source), for instance after waking up from Deep power-down.*

The following steps must be performed to enter Deep power-down mode:

1. Write one to the DPDEN bit in the PCON register
2. (Optional) Set the WAKEUP bit in the PCON register to enable wake-up functionality on PIO0_0
3. Store data to be retained in the general-purpose registers

4. Write one to the SLEEPDEEP bit in the Arm Cortex-M0+ SCR register
5. (Optional) Enable the RTC Timer
6. Use the Arm WFI instruction

Four events can wake the microcontroller from Deep power-down mode:

- RESETN is asserted on the external pin
- An RTC Timer event occurs (if enabled)
- The WAKEUP pin is pulled low externally (if enabled)
- The NFC/RFID core is activated (RFPOW signal)

Any of these event results in a full reset cycle of NHS31xx. If the WAKEUP pin is enabled, the signal should stay stable for 100 µs after the HIGH-to-LOW transition. The following sequence is started:

1. The PMU turns on the on-chip voltage regulators LDO1 and LDO2. When the core voltage reaches the power-on-reset (POR) trip point and the SFRO is stable, a system reset is triggered and the chip reboots
2. If the reason of waking from Deep-power down is different from RESETN or a Watchdog reset, all registers except the GPREG0 to GPREG4, PCON, and RTC registers are in their reset state. If the reason is RESETN or a Watchdog reset, then all registers are in their reset state.
3. Once the chip has booted, read the Deep power-down flag in the PCON register to verify that a wake-up event from Deep power-down caused the reset
4. Establish the cause of exit from Deep-power down by verifying the DPDEXIT bits in the PSTAT register
5. Clear the Deep power-down flag in the PCON register
6. (Optional) Clear the WAKEUP bit in the PCON register
7. Clear the RTC timer event flag in the RTCMIS register
8. (Optional) Read the stored data in the general-purpose registers
9. Set up the PMU for the next Deep power-down cycle

5.4 System power architecture

The NHS31xx platform accepts power from two different sources, from the external power supply pin VDDBAT (domain VBAT) or from the built-in NFC/RFID rectifier (domain VNFC).

VDD_ALON via the VDD_ALON pad powers the external ring (VDDE) of the IO pads, as shown in Figure 8.

Note: *In Passive RF mode, external devices can be powered by connecting them to a PIO0_n (preferably a high-drive pin) set to output logic 1.*

The PMU in the always-on domain then decides on the powering of the internal domains.

The power source decision is as follows:

- If voltage is detected on VBAT and not on VNFC, VBAT powers the internal domains
- If voltage < 1.72 V is detected on VBAT, and higher voltage is detected on VNFC, VNFC powers the internal domains
- If voltage > 1.72 V is detected at both VBAT and VNFC, VBAT powers the internal domains
- Switch-over between sources is possible. For example, if initially both VBAT and VNFC are available, the system is powered from VBAT. If VBAT then becomes unavailable because it is switched off externally or the PSWBAT power switch is overridden, the

internal domains are immediately powered from VNFC. Switch-over is supported in both directions

- When on NFC power only (passive operation), a PROGRAM or ERASE operation on the FLASH memory causes a severe voltage drop, causing a reset. To enable the possibility to PROGRAM, capacitors must be added externally.
  - The use of two capacitors is preferred to lower the serial resistance.
  - Connect each 220 nF capacitor to a GPIO pin, with the other side of the capacitor tied to ground.
  - When the ultra-high drive mode (see Table 252) is enabled, PIO3, PIO7, PIO10, and PIO11 have the lowest resistive values. So, using these pins is preferred.

  In firmware, these capacitors must be put to use according to this sequence, before a PROGRAM operation is started:
  1. Configure both pins as input with pull-up enabled.
     This ensures a slow build-up of the charge in the capacitor.
  2. Wait until the capacitors are almost fully charged.
     The exact time to wait can be calculated in advance. In addition, the pin value can be continually read until it reads a 1. It provides the earliest time possible to move to the next step.
  3. Reconfigure the pins as output.
     When changing the direction, the value that is output on the pin is not taken from the corresponding register. Instead, it is sampled from what was provided when the pin was still an input. If the capacitor is not sufficiently charged, it outputs a 0. If enough time was spent waiting in the second step above, the value becomes a 1. The reconfiguration is necessary to provide a low resistance path between the capacitor and the internal voltage rail. The low resistance path ensures that the charge can be used to overcome the current peak that occurs while flashing.
  4. Explicitly set the pin value to 1. It is not mandatory, but synchronizes the register value with what is actually output on the pin.
- The user can disable the automatic switching by setting bit PMULPM in the PCON register. Setting this bit disables the voltage comparator and forces the selection of the VDDBAT input.

When the system is in Deep Power-down mode, the power source selection is handled in the same way. The powering of the internal domains then depends on the wake-up conditions the user sets: Wake-up due to a 1-to-0 transition on the WAKEUP pin (if WAKEUP is enabled in the PIO0_0 register), an RTC timer event, or an NFC event.

When the WAKEUP function on PIO0_0 is enabled in the PCON register and the DPEN is set, the NHS31xx overrides the user settings of the PIO0_0 pad before entering Deep power-down. The pad is set to wake-up and the pull-up is enabled.

The PSTAT register contains all information on the configuration of the power supplies. The system Power-On Reset reacts on VDD1V2.

Figure 8 shows the power architecture of the NHS31xx. The (analog) IC power controller arbitrates between external battery power (VBAT) and RF power (VNFC) and controls the main power switches. The Power management unit controls the LDOs and the power flow to the different circuits block inside the NHS31xx, as well as the different sleep modes. It interacts with the RTC Timer (see Section 4.5.16) and the Arm Cortex-M0+ core. The RTC timer resides partly in the always-on power domain, partly in the 1.2 V domain. It generates a power-on request signal toward the PMU.

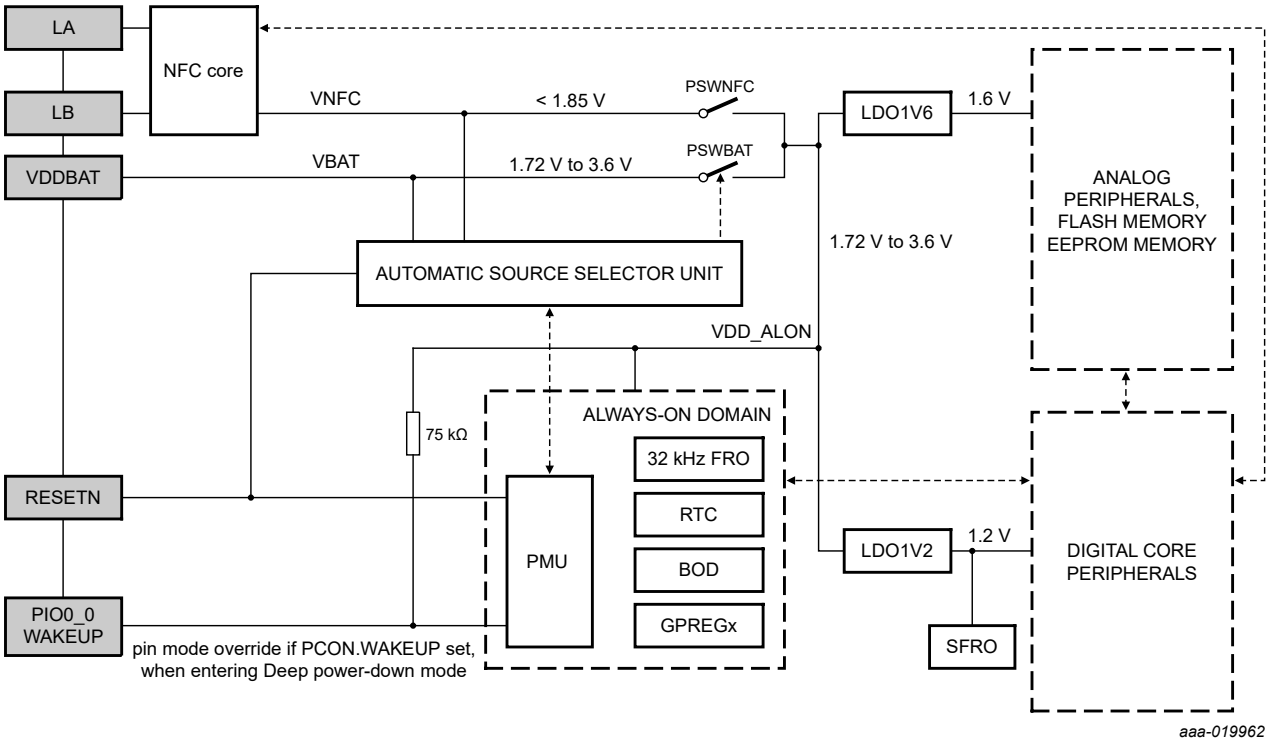


Figure 8. NHS31xx power architecture

In Figure 8, PSWNFC and PSWBAT are power switches. LDO1V2 converts voltages in the range 1.72 V to 3.6 V into 1.22 V. LDO1V6 converts voltages in the range 1.72 V to 3.6 V into 1.6 V. Each LDO can be enabled separately. The always-on power domain is automatically switched to any available power via the autoswitch unit.

A 1.2 nF buffer capacitor is included at the input of the LDOs to maintain system integrity during Miller pauses. Figure 9 shows the different power supply pads and power supply connections used in the design. The main power inputs are via VDD_ALON/VSS pads, connecting to the VDDE and GNDE rails. Switched battery power is connected via a VDDBAT pad.

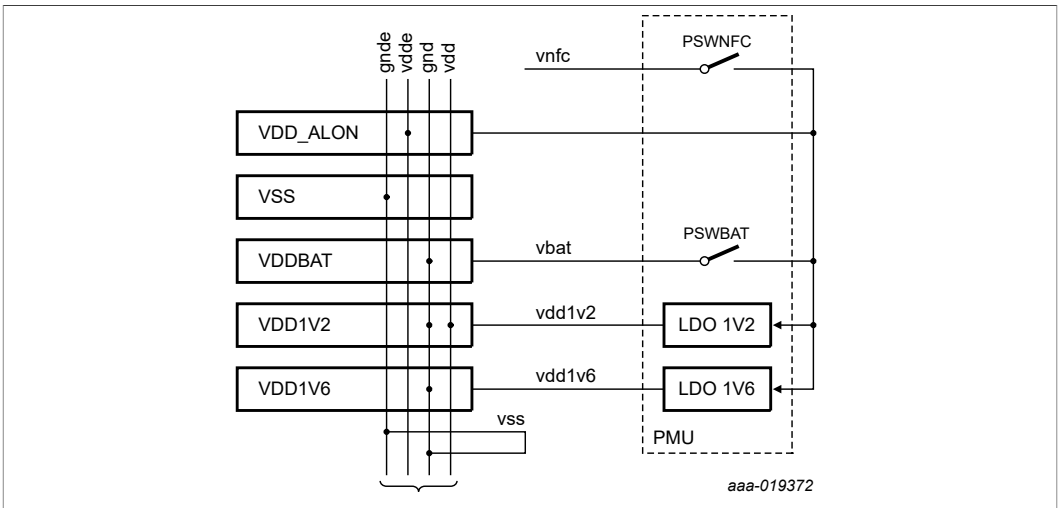


Figure 9. Power rails and power supply pad types

The PMU states and settings of the LDOs are summarized in Table 32, and the state transitions are shown in Figure 10.

Table 32. IC power states

State	VDD_ALON	DPDN[1]	DSLP[2]	LDO1 1.2 V	LDO2 1.6 V
NOPOWER	no	X	X	X	X
ACTIVE	yes	0	0	on	on
DEEPPDN	yes [3]	1	0	off	off
DEEPSLEEP	yes [3]	0	1	on	off

[1] DPDN indicates whether the system is in Deep power-down.
[2] DSLP indicates that the system is in Deep-sleep.
[3] The value of VDD_RFID is X instead of logic 0 if the RFID interrupt is disabled in the PMU settings.

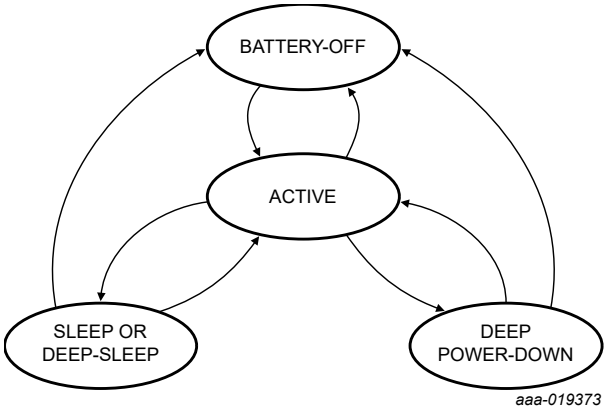


Figure 10. PMU state transition diagram

To avoid deadlock situations, the RESETN signal also forces a 'request power-on' when in Deep power-down or Deep-sleep. Table 33 and Table 34 summarize the events that can influence wake-up from Deep-sleep or Deep power-down (DEEPPDN or DEEPSLEEP to ACTIVE state transition).

Table 33. State transition events for DEEPSLEEP to ACTIVE

Event	Description
RESETN	reset asserted
RTC event	if the timer reaches pre-set value
Watchdog	watchdog issues interrupt or reset
WAKEUP	signal on WAKEUP pin or pins defined for exit out of Deep-sleep
RF field	RF field is detected, potential NFC command input (if set in PMU)
Start logic interrupt	one of the enabled start logic interrupts is asserted

Table 34. State transition events for DEEP POWER-DOWN to ACTIVE

Event	Description
RESETN	reset asserted
RTC event	if the timer reaches pre-set value
WAKEUP	signal on WAKEUP pin
RF field	RF field is detected, potential NFC command input (if set in PMU)

The power-up sequence is shown in Figure 11. When external power is applied and the PSUBAT switch closed, the always-on part gets a power-on reset signal and the timer FRO starts running. The TFRO starts a small state machine in the PMU. In the first state, the LDO powering the digital domain is started. In the second state, the LDO powering the 1.6 V analog domain is started. In the last state, the system is considered 'on'. The transition from 'off' to 'on' takes approximately 92 µs. In the VDD1V2 domain, enabling the LDO1 and SFRO triggers the system_por. The system can boot when the Flash memory is operational.

If there is no external power, but there is RF power, the same procedure is followed except that PSWNFC connects power to the LDOs.

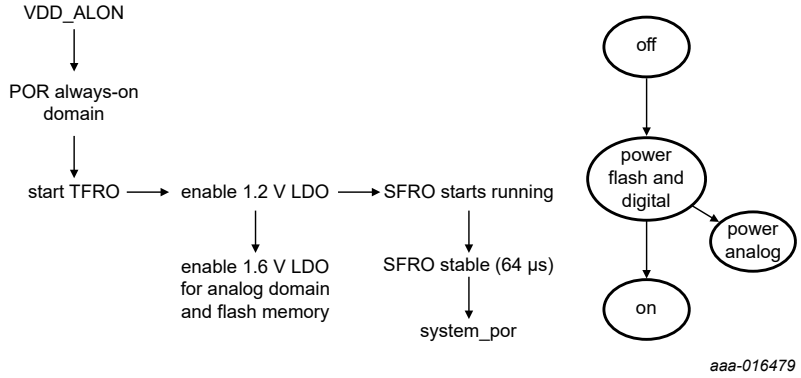


Figure 11. NHS31xx power-up sequence

The user cannot disable the TFRO as the PMU uses it.

5.5 Brownout detection

The NHS31xx monitors the voltage on VDD_ALON. If this voltage falls below the threshold level (1.8 V), the BOD asserts an interrupt.

The interrupt detection has a hysteresis of 75 mV.

5.6 Register descriptions

All PMU registers require a synchronized access (see Section 5.6.4).

Table 35. Register overview: PMU (base address 0x4003 8000)

Name	Access	Address offset	Description	Reset value
PCON	R/W	0x000	power control register	0x0000 0000
GPREG0	R/W	0x004	general-purpose register 0	0x0000 0000
GPREG1	R/W	0x008	general-purpose register 1	0x0000 0000
GPREG2	R/W	0x00C	general-purpose register 2	0x0000 0000
GPREG3	R/W	0x010	general-purpose register 3	0x0000 0000
GPREG4	R/W	0x014	general-purpose register 4	0x0000 0000
PSTAT	R	0x018	power management unit status register	n/a
-	-	0x01C	reserved	-
ACCSTAT	R	0x020	access status register	0x0000 0001
LDO1V6	R/W	0x024	analog 1.6 V LDO trimming	0x0000 0004
TMRCLKCTRL	R/W	0x02C	timer clock control register	0x0060 0001
IMSC	R/W	0x030	interrupt mask set and clear Register	0x0000 0000
RIS	R	0x034	raw interrupt status register	0x0000 0000
MIS	R	0x038	masked interrupt status register	0x0000 0000
ICR	W	0x03C	interrupt clear register	n/a

5.6.1 Power control register (PCON)

The power control register selects whether one of the ARM Cortex-M0+ controlled Power-down modes (Sleep mode or Deep-sleep mode) or the Deep power-down mode is entered. The register also provides the flags for Sleep or Deep-sleep modes and Deep power-down modes respectively. It also contains the overrides for the power source selection.

Table 36. PCON register (address 0x4003 8000) bit description

Bit	Symbol	Description		Reset value
0	-	reserved. Do not write 1 to this bit.		0
1	DPEN	Deep power-down mode enable		0
		0	ARM WFI enters Sleep or Deep-sleep mode (clock to ARM Cortex-M0+ core turned off)	
		1	ARM WFI enters Deep-power down mode (ARM Cortex-M0+ core powered down)	
7:2	-	reserved. Do not write 1 to this bit.		0

Table 36. PCON register (address 0x4003 8000) bit description *...continued*

Bit	Symbol	Description		Reset value
8	SLEEPFLAG	Sleep mode flag		0
		0	**read:** No Power-down mode entered **write:** No effect	
		1	**read:** Sleep/Deep-sleep mode entered **write:** write logic 1 to clear SLEEPFLAG	
10:9	-	reserved. Do not write 1 to these bits.		0
11	DPDFLAG	Deep power-down flag		0
		0	**read:** No power-down mode entered **write:** No effect	
		1	**read:** Deep power-down mode entered **write:** Clear the deep power-down flag	
12	-	reserved. Do not write 1 to this bit.		0
13	PMULPM [1]	PMU ultra-low-power mode		0
		0	power switches are in automatic mode	
		1	disable automatic switching (default to external power)	
14	VBAT	Battery switch control		0
		0	Automatic mode	
		1	force off (disconnect external battery)	
15	BODEN	Brown-Out Detector enable		0
		0	BOD disabled	
		1	BOD enabled	
16	-	reserved. Do not write 1 to this bit.		0
18:17	FORCEVBAT FORCEVNFC	force the power source selection		0
		01	force to VBAT/external power	
		10	force to NFC power	
		11	autoswitching if PMULPM is 0	
19	WAKEUP [2]	enable the WAKEUP function on pin PIO0_0		-
		0	WAKEUP function not enabled	
		1	WAKEUP function assigned to PIO0_0. When selecting this option, PIO0_0 has a weak pull-up to VDD_ALON when in Deep power-down.	
31:20	-	reserved. Do not write 1 to this bit.		-

[1] Setting the PMULPM bit disables the automatic switching. It is recommended only when on battery power and no switching to NFC is foreseen.

[2] Setting the WAKEUP bit enables the wake-up function of the PIO0_0 pad. Upon entering Deep power-down, the pad is automatically set to input with disabled receiver, and enabled pull-up resistor. If this bit is not set, it is not possible to exit Deep power-down via the PIO0_0 pin.

5.6.2 General-purpose registers 0 to 4 (GPREG0-4)

The general-purpose registers retain data through Deep power-down mode when power is still applied to the VDDBAT pin but the chip has entered Deep power-down mode. Only a cold boot when all external and RF power has been removed from the chip, or a hard reset (asserting the RESETN pin), resets the general-purpose registers.

Table 37. GPREG0 - GPREG4 registers (address 0x4003 8004 to 0x4003 8014) bit description

Bit	Symbol	Value	Description	Reset value
31:0	GPDATA	-	data retained during Deep power-down mode	0

5.6.3 Power status register (PSTAT)

The power status register indicates the power source and configuration of LDOs and power switches. Reading the PSTAT register clears the BOD and RFPOW interrupts.

Table 38. PSTAT register (address 0x4003 8018) bit description

Bit	Symbol	Description		Reset value
0	PSWNFC	status of NFC power selection switch		1 or 0
		0	switch disabled (not conducting)	
		1	switch enabled (conducting)	
1	PSWBAT	status of battery power selection switch		0 or 1
		0	switch disabled (not conducting)	
		1	switch enabled (conducting)	
2	-	reserved		0 or 1
4:3	DPDEXIT	reason for exit from Deep-Power Down. Only valid when the DPDFLAG in the PCON register is asserted (1)		0
		00	POR or RESETN pin asserted	
		01	RTC timer event	
		10	RF field detected	
		11	WAKEUP pin negative edge detected	
5	BOD1V8[1]	BOD detected (VDD < 1.8 V)		0
		0	VDD ≥ 1.8 V	
		1	VDD < 1.8 V	
6	-	reserved		
7	RFPOW [1]	RF field detected		0
		0	no field	
		1	RF field	
31:8		reserved		-

[1] The BOD1V8 and RFPOW signals are real-time (immediate) values, and might differ from the corresponding interrupts

5.6.4 Access status register (ACCSTAT)

The access status register indicates if the PMU is able to accept data over the APB. The PMU runs at a different clock rate than the core system. So, bus synchronization is necessary. This mechanism is described below.

Write access:

The write procedure is as follows:

- When the Arm writes to a PMU register:
  - The address/value pair is stored in the command-buffer
  - The direction flag is set to write
  - ACCSTAT goes to logic 0 indicating a command is pending. This transition happens on an edge of the APB clock, which is faster than the TFRO clock.
- When the TFRO clock edge comes, the address/value in the command-buffer is written to the RTC block
- On a next TFRO edge, the ACCSTAT goes to logic 1 to indicate that no command is pending and a new one can be given

Read access:

The read procedure is as follows:

- When a value is stored in the PMU register for the Arm to read:
  - Only the register address is stored in the command-buffer
  - The direction flag is set to read
  - ACCSTAT goes to logic 0 indicating a command is pending.
- When the TFRO clock edge arrives, the address in the buffer is used to read the actual RTC register. This register value is copied in the data part of the command-buffer
- On the next TFRO edge, the ACCSTAT goes to logic 1 to indicate that the command is completed
- The Arm must do another read from a PMU register. This read ensures that the actual register value is picked up from that data part of the command-buffer
- The command-buffer sees that the address of the second read is the same as the address part in the command-buffer. In this case, the result is not treated as a new read command (ACCSTAT stays logic 1)
- However, the address part of the command-buffer is now cleared, so that a third read (even when from the same PMU register) causes a new command

Table 39. ACCSTAT register (address 0x4003 8020) bit description

Bit	Symbol	Description	Reset value
0	PMU_READY	logic 0: PMU access not possible, logic 1: PMU access possible	1
31:1	-	reserved	-

5.6.5 Timer clock control register (TMRCLKCTRL)

The timer clock control register controls the 32 kHz oscillator and RTC clock source.

Note: *This register resides in the always-on power domain, hence its location in the PMU. The 32 kHz TFRO is always running when VDD_ALON is available.*

Table 40. TMRCLKCTRL register (address 0x4003 802C) bit description

Bit	Symbol	Description		After boot
0	TMREN	RTC clock source.		1
		0	no clock	
		1	32 kHz FRO	
15:1	-	reserved		-
21:16	TMRCLKTRIM	timer FRO trim value (6-bit)		device dependent
23:22	TMRCURRLVL	timer FRO current level selector bits		device dependent
31:24	-	reserved		-

5.6.6 PMU interrupt mask set and clear register (IMSC)

This register controls if each of the two possible interrupt conditions in the PMU is enabled.

Table 41. IMSC register (address 0x4003 8030) bit description

Bit	Symbol	Description	Reset value
0	BODI	software must set this bit to enable an interrupt when the BOD is triggered	0
1	RFPOWI	software must set this bit to enable an interrupt when RF power is detected	0
31:2	-	reserved	0

5.6.7 PMU raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless if the interrupt is enabled in the IMSC register.

Table 42. RIS register (address 0x4003 8034) bit description

Bit	Symbol	Description	Reset value
0	BODRI	raw BOD interrupt value	0
1	RFPOWRI	raw RF power interrupt value	0
31:2	-	reserved	0

5.6.8 PMU masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the IMSC registers.

Table 43. MIS register (address 0x4003 8038) bit description

Bit	Symbol	Description	Reset value
0	BODRI	raw BOD interrupt value	0
1	RFPOWRI	raw RF power interrupt value	0
31:2	-	reserved	0

5.6.9 PMU interrupt clear register (ICR)

To clear the corresponding interrupt condition or conditions in the PMU, software can write one or more logic 1s to this write-only register.

Note: *Clearing the corresponding bit in IMSC registers disables the interrupt conditions.*

Table 44. ICR register (address 0x4003 803C) bit description

Bit	Symbol	Description	Reset value
0	BODC	writing a logic 1 clears the BOD interrupt	0
1	RFPOWC	writing a logic 1 clears the RF power interrupt	0
31:2	-	reserved	0

6 In-application programming (IAP) firmware

The in-application programming (IAP) provides APIs to program a blank device. The APIs can also erase and reprogram a previously programmed device or program the flash memory using the application program in a running system.

The following functions are supported:

- Return part identity
- Return boot code version
- Return device serial number UID
- Prepare/unlock a Flash sector for program/erase
- Erase one or more contiguous Flash sectors based on sector range
- Copy data to Flash
- Write Flash sector

The in-application programming (IAP) library is stored, together with the boot code, in read-only (locked) sectors in the Flash memory space.

The IAP routine must be called with a word pointer in register R0 pointing to memory (RAM) containing command code and parameters. Result of the IAP command is returned in the result table register R1 points to. Figure 12 illustrates the IAP command passing procedure.

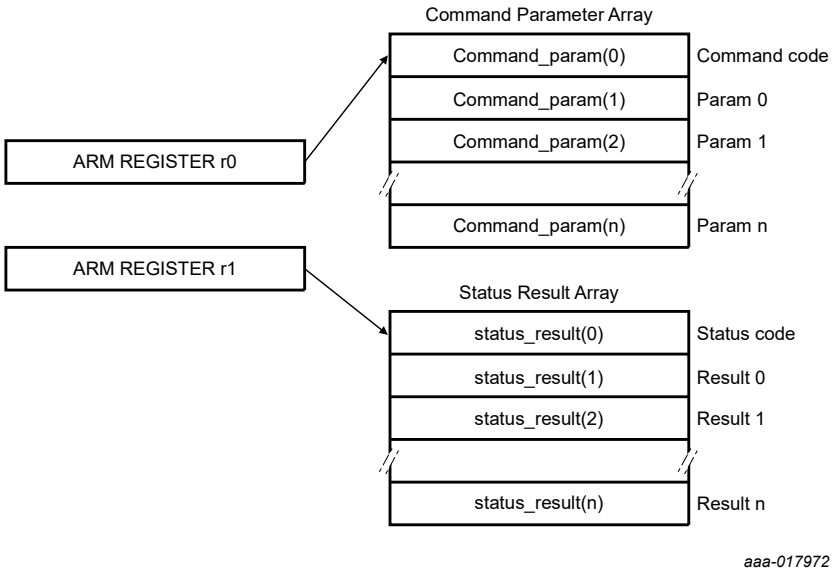


Figure 12. IAP command passing

The number of parameters and results vary according to the IAP command. The maximum number of parameters is 5, passed to the 'Copy Data to Flash' command. The maximum number of results is 4, returned by the 'Read UID' command. When an undefined command is received, the command handler sends the status code INVALID_COMMAND. The IAP routine resides at the 0x1FFF 1FF0 location and it is thumb code. Call the IAP function in the following way:

Step 1: Define the IAP location entry point. To indicate that it is a thumb instruction at the destination address, bit 0 of the IAP location is set.

```
#define IAP_LOCATION 0x1fff1ff1
```

Step 2: To pass IAP command table and result table to the IAP function, define data structure or pointers:

```
unsigned int command_param[5];
unsigned int status_result[4];
```

or

```
unsigned int * command_param;
unsigned int * status_result;
command_param = (unsigned int *) 0x...
status_result = (unsigned int *) 0x...
```

Step 3: Define function type pointer, which takes two parameters and returns void. The IAP returns the result with the base address of the table residing in R1.

```
typedef void (*IAP)(unsigned int [], unsigned int[]);
IAP iap_entry;
```

Setting the function pointer:

```
iap_entry = (IAP) IAP_LOCATION;
```

Step 4: To call the IAP, use the following statement:

```
iap_entry (command_param, status_result);
```

The flash memory is not accessible during program/erase operations. IAP commands, which result in a flash write/erase operation, use 16 bytes of space in the top portion of the on-chip RAM for execution. This portion is initialized at boot. If the user application executes IAP calls for flash programming, this area must be left untouched.

Many of the IAP commands access non-volatile memories. And may cause a temporary power consumption increase.

6.1 IAP command list

Table 45. IAP command list

IAP Command	Command code		Reference
	Hex	Decimal	
Read factory settings	28h	40d	Section 6.1.1
Prepare sectors for write operation	32h	50d	Section 6.1.2
Copy data to flash	33h	51d	Section 6.1.3
Erase sectors	34h	52d	Section 6.1.4
Blank check sectors	35h	53d	Section 6.1.5
Read part identity	36h	54d	Section 6.1.6

Table 45. IAP command list...*continued*

IAP Command	Command code		Reference
	Hex	Decimal	
Read boot code version	37h	55d	Section 6.1.7
Compare	38h	56d	Section 6.1.8
Read UID	3Ah	58d	Section 6.1.9
Erase page	3Bh	59d	Section 6.1.10

After power-on reset, all flash sectors are protected. To erase or write to sectors or pages, the user first must issue the 'Prepare sectors for write operation' command to unprotect the corresponding sectors.

For example, if the user wants to write data to sector 6, the following procedure must be followed:

1. Call the 'Prepare' command (code: 50d) for sector 6.
2. Call the 'Erase' command (code: 52d) to erase sector 6.
3. Call the 'Prepare' command (code: 50d) again.
4. Call the 'Copy Data to Flash' command (code: 51d) to write data to sector 6.

Note: *During an erase or program operation, the flash is not accessible. So, if the interrupt vector table is placed in flash (default setup), the application must ensure that no interrupts are triggered. Accessing the table, if in flash, results in a hard fault.*

6.1.1 Read factory settings

This command reads the factory settings for calibration registers.

Table 46. IAP Read factory settings command

Command	Read factory settings
Input	command code: 40 (decimal) param0: The address of the targeted calibration register. See Table 44 for supported registers
Status code	CMD_SUCCESS | PARAMETER_ERROR
Result	result0: The factory value when the status code is CMD_SUCCESS. PARAMETER_ERROR is returned when param0 is not one of the supported calibration registers
Description	This command provides users with the capability to read the factory value for the calibration registers

Table 47. Supported registers for factory setting read

Parameter 0	Description
0x4006 0010 [1]	TSENSP1: Temperature sensor setup register 1
0x4006 0014 [1]	TSENSP2: Temperature sensor setup register 2
0x4006 0018 [1]	TSENSP3: Temperature sensor setup register 3
0x4005 4008	RTCCAL: RTC timer calibration register

[1] Only applicable to devices that support temperature sensors

6.1.2 Prepare sectors for write operation

This command makes the sector or sectors unprotected.

Table 48. IAP Prepare sectors for write operation command

Command	Prepare sector for write operation
Input	command code: 50 (decimal) param0: Start Sector Number param1: End Sector Number (must be greater than or equal to start sector number)
Status code	CMD_SUCCESS | INVALID_SECTOR
Result	none
Description	This command must be executed before executing the 'Copy Data to flash' or the 'Erase Sectors' command. Successful execution of the 'Copy Data to flash' or 'Erase Sectors' command causes relevant sectors to be protected again. This command cannot be used to prepare the boot sector. To prepare a single sector, use the same 'Start' and 'End' sector numbers

6.1.3 Copy data to flash

This command copies the data from the RAM buffer/EEPROM to the specified flash area to be written.

Table 49. IAP Copy data to flash command

Command	Copy data to flash
Input	command code: 51 (decimal) param0 (DST): Destination flash write address. This address must be a 64 byte boundary. param1 (SRC): Source read address (RAM / EEPROM). This address must be a word boundary. param2: Number of bytes to be written. Must be a multiple of 64. param3: System clock frequency (CCLK) in kHz. This parameter is ignored.[1]

Table 49. IAP Copy data to flash command*...continued*

Command	Copy data to flash
Status code	CMD_SUCCESS | SRC_ADDR_ERROR (Address not a word boundary) | DST_ADDR_ERROR (Address not on correct boundary) | VDST_ADDR_NOT_VALID | VSRC_ADDR_NOT_VALID | COUNT_ERROR (Byte count is not a multiple of the page size) | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION
Result	none
Description	This command is used to program the flash memory. The affected sectors must be prepared first by calling 'Prepare Sector for Write Operation' command. The flash area to be written must be blank (or cleared) before the operation is called. The affected sectors are automatically protected again once the copy command is successfully executed. This command cannot be used to write to the boot sector

[1] These devices are not able to select external clocks.

6.1.4 Erase sectors

Table 50. IAP Erase sector command

Command	Erase Sector
Input	command code: 52 (decimal). param0: Start Sector Number. param1: End Sector Number (must be greater than or equal to start sector number). param2: System clock frequency (CCLK) in kHz. This parameter is ignored.[1]
Status code	CMD_SUCCESS | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | INVALID_SECTOR
Result	none
Description	This command is used to erase a sector or multiple sectors of on-chip flash memory. The affected sectors must be prepared first by calling the 'Prepare sector for write operation' command. This command cannot be used to erase the boot sector. To erase a single sector, use the same 'Start' and 'End' sector numbers

[1] These devices are not able to select external clocks.

6.1.5 Blank-checking sectors

Table 51. IAP Blank-check sector command

Command	Blank check sector
Input	command code: 53 (decimal) param0: Start Sector Number param1: End Sector Number (must be greater than or equal to start sector number)

Table 51. IAP Blank-check sector command*...continued*

Command	Blank check sector
Status code	CMD_SUCCESS | SECTOR_NOT_BLANK | INVALID_SECTOR
Result	result0: Offset of the first non-blank word location if the Status Code is SECTOR_NOT_BLANK. result1: Contents at the non-blank word location
Description	This command is used to blank-check a sector or multiple sectors of on-chip flash memory. To blank-check a single sector, use the same 'Start' and 'End' sector numbers

6.1.6 Read part identification number

Table 52. IAP Read part identification command

Command	Read part identification number
Input	command code: 54 (decimal) parameters: None
Status code	CMD_SUCCESS
Result	result 0: Part Identification Number
Description	This command is used to read the part identification number.

6.1.7 Read boot code version number

Table 53. IAP Read boot code version number command

Command	Read boot code version number
Input	command code: 55 (decimal) parameters: None
Status code	CMD_SUCCESS
Result[1]	result 0: 2 bytes of boot code version number. Read as <byte 1 (major)><byte 0 (minor)>
Description	This command is used to read the boot code version number.

[1] Ignore the data contained in byte 2 and byte 3 in Result 0, which is for internal debugging purposes only.

6.1.8 Compare

Table 54. IAP Compare command

Command	Compare
Input	command code: 56 (decimal) param0 (DST): Starting flash, RAM or EEPROM address of data bytes to be compared. This address must be a word boundary. param1 (SRC): Starting flash, RAM or EEPROM address of data bytes to be compared. This address must be a word boundary. param2: Number of bytes to be compared and must be a multiple of 4.

Table 54. IAP Compare command*...continued*

Command	Compare
Status code	CMD_SUCCESS | COMPARE_ERROR | COUNT_ERROR (Byte count is not a multiple of 4) | ADDR_ERROR
Result	result0: If the Status Code is COMPARE_ERROR, the first mismatch is offset.

6.1.9 Read UID

Table 55. IAP Read Unique Identification number command

Command	Read UID
Input	command code: 58 (decimal) parameters: None
Status code	CMD_SUCCESS
Result	result 0: The first 32-bit word result 1: The second 32-bit word result 2: The third 32-bit word result 3: The fourth 32-bit word
Description	This command is used to read the unique ID.

6.1.10 Erase page

Table 56. IAP Erase page command

Command	Erase Page
Input	command code: 59 (decimal) param 0: Start page number param 1: End page number (must be greater than or equal to start page). param 2: System clock frequency (CCLK) in kHz. This parameter is ignored.[1]
Status code	CMD_SUCCESS | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | INVALID_SECTOR
Result	none
Description	This command is used to erase a page or multiple pages of on-chip flash memory. The affected sectors must be prepared first by calling 'Prepare Sector for Write Operation' command. To erase a single page, use the same 'start' and 'end' page number.

[1] These devices are unable to select external clocks.

6.2 IAP status codes

Table 57. IAP Status Codes

Status code	Mnemonic	Description
0	CMD_SUCCESS	command is executed successfully
1	INVALID_COMMAND	invalid command
2	SRC_ADDR_ERROR	source address is not on a word boundary
3	DST_ADDR_ERROR	destination address is not on a correct boundary
4	SRC_ADDR_NOT_MAPPED	source address is not mapped in the memory map. Count value is considered where applicable.
5	DST_ADDR_NOT_MAPPED	destination address is not mapped in the memory map. Count value is considered where applicable.
6	COUNT_ERROR	byte count is not a permitted value
7	INVALID_SECTOR	sector number is invalid
8	SECTOR_NOT_BLANK	sector is not blank
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	command to prepare sector for write operation was not executed
10	COMPARE_ERROR	source and destination data are not the same
11	-	this error code is not used
12	PARAMETER_ERROR	invalid parameter or Insufficient number of parameters
13	ADDR_ERROR	address is not on a word boundary

7 Temperature sensor

7.1 About this section

This section describes the use of the temperature sensing feature of the NHS31xx family of sensor node ICs.

7.2 Introduction

The temperature sensor is integrated into the chip and can accurately measure temperatures over a wide range.

7.3 Features

The temperature sensor block measures the temperature of the die and outputs a calibrated value in Kelvin. It has:

- ±0.5 °C absolute temperature accuracy between −40 °C and 0 °C
- ±0.3 °C absolute temperature accuracy between 0 °C and +45 °C
- ±0.5 °C absolute temperature accuracy between +45 °C and +85 °C

7.4 General description

Register base address of the temperature sensor block: 0x4006 0000.

The temperature is measured using a high-precision zoom-ADC.

Table 58. Conversion time for different resolution of TSENS

Resolution (bit)	Resolution (°C)	Conversion time (ms)
7	± 0.8	4
8	± 0.4	7
9	± 0.2	14
10	± 0.1	26
11	± 0.05	50
12	± 0.025	100

7.5 Calibration

The temperature sensor calibration values are determined during manufacturing. For increased accuracy, the user can provide custom parameters A, B, and alpha. For calibration, the raw value X is needed, and high accuracy measurement is recommended. Internally, the following equation is used to obtain the calibrated temperature output:

$$T = A \times \frac{\alpha}{\alpha + X} + B \quad (1)$$

The calibration values A, B, and alpha for the default resolution settings are determined during the calibration phase of the manufacturing process and stored in EEPROM.

For detailed information on calibrating the temperature sensor, see the NHS31xx temperature sensor calibration application note (Ref. 5).

7.6 Register descriptions

The register addresses for the temperature sensor are listed in Table 59.

Table 59. Register overview for temperature sensor (base address 0x4006 0000)

Name	Access	Address offset	Description	Reset value [1]
CR	R/W	0x0000	control register	0
DR	R	0x0004	data register. Contains result of conversion	0
SR	R	0x0008	status register	0x0000 0000
SP0	R/W	0x000C	setup register 0: set ADC and operation modes	0x0000 000F
SP1	R/W	0x0010	setup register 1: Calibration value ‘A’	0
SP2	R/W	0x0014	setup register 2: Calibration value ‘B’	0
SP3	R/W	0x0018	setup register 3: Calibration value ’alpha’	0
TLO	R/W	0x001C	low temperature threshold register	0x0000 8000
THI	R/W	0x0020	high temperature threshold register	0x0000 7FFF
IMSC	R/W	0x0024	interrupt mask set and clear register	0
RIS	R	0x0028	raw interrupt status register	0
MIS	R	0x002C	masked interrupt status register	0
ICR	W	0x0030	interrupt clear register	n/a

[1] Reset value reflects the data stored in used bits only. It is not valid for the content of reserved bits.

7.6.1 Temperature sensor control register (CR)

Table 60. CR register (address 0x4006 0000) bit description

Bit	Symbol	Description	Reset value	Access
0	START	sensor start bit. The temperature sensor clears this bit at the end of the measurement.	0	R/W
31:1	-	reserved	-	-

7.6.2 Temperature sensor data register (DR)

The DR register holds the current temperature data in either raw or calibrated format. Reading the DR register clears the temperature sensor interrupts.

Table 61. DR register (address 0x4006 0004) bit description

Bit	Symbol	Description	Reset value	Access
15:0	TDATA	result of last conversion. Format depends on the TOUTMODE bit of the SP0 register. Raw format is unsigned fixed point (5.11); calibrated format is signed fixed point (10.6) 2-complement	0x0000	R
31:16	-	reserved	-	-

7.6.3 Temperature sensor status register (SR)

The SR register holds the status of the temperature sensor.

Table 62. SR register (address 0x4006 0008) bit description

Bit	Symbol	Description	Reset value	Access
0	TRANLOC	coarse range. Logic 1: out of range (too low)	0	R
1	TRANHIC	coarse range. Logic 1: out of range (too high)	0	R
2	TRANLOF	fine range. Logic 1: out of range (too low)	0	R
3	TRANHIF	fine range. Logic 1: out of range (too large)	0	R
4	TSUCC	logic 1: successful conversion	0	R
7:5	TRESMODE	used Resolution mode	0	R
8	TOUTMODE	used Output mode	0	R
31:16	-	reserved	-	-

7.6.4 Temperature sensor setup register 0 (SP0)

The SP0 register controls the resolution and mode settings for the zoom-ADC of the temperature sensor.

Table 63. SP0 register (address 0x4006 000C) bit description

Bit	Symbol	Description		Reset value	Access
0	TOUTMODE	temperature sensor Output mode [0]:		1	R/W
		0	raw data		
		1	calibrated data		
3:1	TRESMODE	temperature sensor Resolution mode:		111b	R/W
		010b	7-bit mode		
		011b	8-bit mode		
		100b	9-bit mode		
		101b	10-bit mode		
		110b	11-bit mode		
		111b	12-bit mode		
5:4	-		reserved	-	-
6	TVCALEN		temperature sensor voltage calibration enable[1]	0	R/W
31:7	-		reserved	-	-

[1] See Ref. 5 for information on calibrating the temperature sensor

7.6.5 Temperature sensor setup register 1 (SP1)

The SP1 register contains the calibration parameter ‘A’.

Table 64. SP1 register (address 0x4006 0010) bit description

Bit	Symbol	Description	Reset value	Access
15:0	A	calibration factor ‘A’ unsigned fixed point (10.6) 2-complement	0x0000	R/W
31:16	-	reserved	-	-

7.6.6 Temperature sensor setup register 2 (SP2)

The SP2 register contains the calibration parameter ‘B’.

Table 65. SP2 register (address 0x4006 0014) bit description

Bit	Symbol	Description	Reset value	Access
15:0	B	calibration factor ‘B’ signed fixed point (10.6) 2-complement	0x0000	R/W
31:16	-	reserved	-	-

7.6.7 Temperature sensor setup register 3 (SP2)

The SP3 register contains the calibration parameter ‘alpha’.

Table 66. SP3 register (address 0x4006 0018) bit description

Bit	Symbol	Description	Reset value	Access
15:0	ALPHA	calibration factor ‘alpha’ unsigned fixed point (6.10) 2-complement	0x0000	R/W
31:16	-	reserved	-	-

7.6.8 Temperature sensor low-temperature threshold register (TLO)

The TLO register contains the low-temperature threshold value, which is compared to the measured (calibrated) temperature. If the measured temperature is lower than the value in TLO and the corresponding bit in the IMSC register is set, then an interrupt is generated.

Table 67. TLO register (address 0x4006 001C) bit description

Bit	Symbol	Description	Reset value	Access
15:0	TLO	low-temperature threshold value in signed fixed point (10.6) 2-complement	0x8000	R/W
31:16	-	reserved	-	-

7.6.9 Temperature sensor high-temperature threshold register (THI)

The THI register contains the high-temperature threshold value, which is compared to the measured (calibrated) temperature. If the measured temperature is higher than the value in THI and the corresponding bit in the IMSC register is set, then an interrupt is generated.

Table 68. THI register (address 0x4006 0020) bit description

Bit	Symbol	Description	Reset value	Access
15:0	THI	high-temperature threshold value in signed fixed point (10.6) 2-complement	0x7FFF	R/W
31:16	-	reserved	-	-

7.6.10 Temperature sensor interrupt mask set/clear register (IMSC)

This register controls which interrupts are enabled.

Table 69. IMSC register (address 0x4006 0024) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	software must set this bit to enable interrupt when a temperature conversion is ready on DR	0	R/W
1	TLOE	software must set this bit to enable interrupt when a temperature conversion is lower than the value stored in TLO	0	R/W
2	THIE	software must set this bit to enable interrupt when a temperature conversion is higher than the value stored in THI	0	R/W
31:3	-	reserved	-	-

7.6.11 Temperature sensor raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless of whether the interrupt is enabled in the IMSC register.

Table 70. RIS register (address 0x4006 0028) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	this bit is logic 1 when a temperature conversion has finished	0	R
1	LOI	software must set this bit to enable interrupt when a temperature conversion is below the value stored in TLO	0	R
2	HII	this bit is logic 1 when a temperature conversion has finished and the value exceeds the THI threshold value	0	R
31:3	-	reserved	-	-

7.6.12 Temperature sensor register masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the IMSC registers. When a temperature sensor interrupt occurs, the interrupt handler must read this register to determine the cause or causes of the interrupt.

Table 71. MIS register (address 0x4006 002C) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	this bit is logic 1 when a temperature conversion has finished and this interrupt is enabled	0	R
1	LOI	this bit is logic 1 when a temperature conversion has finished, if the value is below the LOI threshold value, and this interrupt is enabled	0	R
2	HII	this bit is logic 1 when a temperature conversion has finished, if the value exceeds the THI threshold value, and this interrupt is enabled	0	R
31:3	-	reserved	-	-

7.6.13 Temperature sensor register interrupt clear register (ICR)

Software can write one or more ones to this write-only register, to clear the corresponding interrupt condition or conditions in the temperature sensor controller.

Note: *The interrupt conditions can be cleared by reading the DR register or disabled by clearing the corresponding bit in IMSC registers.*

Table 72. ICR register (address 0x4006 0030) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYC	writing a logic 1 clears the temperature sensor conversion ready interrupt	0	W
1	TLOC	writing a logic 1 clears the temperature sensor low threshold interrupt	0	W
2	THIC	writing a logic 1 clears the temperature sensor high threshold interrupt	0	W
31:3	-	reserved	-	-

8 Current-to-digital converter (I2D)

8.1 About this chapter

This chapter describes the setup and use of the current-to-digital converter

8.2 Introduction

One of the available components of the NHS31xx family is a Current-to-Digital converter. This Current-to-Digital converter is based on a 16-bit I/F converter with selectable integration time.

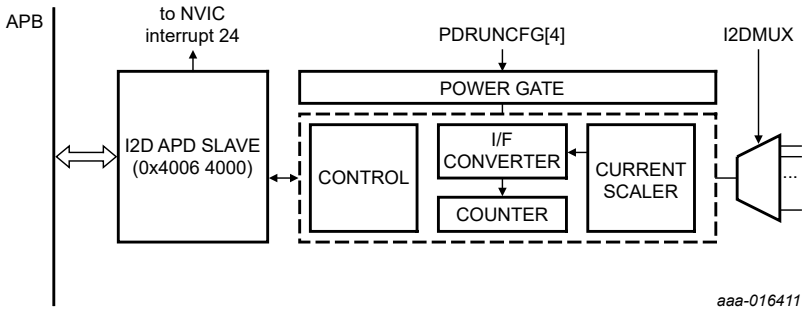


Figure 13. Current-to-Digital block diagram

8.3 Input multiplexer

The input to the Current-to-Digital converter is connected to the analog buses via an analog multiplexer. Table 73 shows the inputs of the multiplexer.

Table 73. Connections to the Current-to-Digital analog input multiplexer

AMUX input	Source	Description
0	ana_extbus0	(external) pin ANA0_0
1	ana_extbus1	(external) pin ANA0_1
2	ana_extbus2	(external) pin ANA0_2
3	ana_extbus3	(external) pin ANA0_3
4	ana_extbus4	(external) pin ANA0_4
5	ana_extbus5	(external) pin ANA0_5

8.4 ENOB

The different input modes have different INL (Integral Non-Linearity) due to the different preamplifiers used for input ranging. The following table shows the ENOB (Effective Number Of Bits) for each range.

Table 74. ENOB for different input ranges and corrections

Input range	Offset/gain corrected (2-point calibration)`	Quadratic corrected (3-point calibration)
5 nA	7	8
25 nA	9	9.3

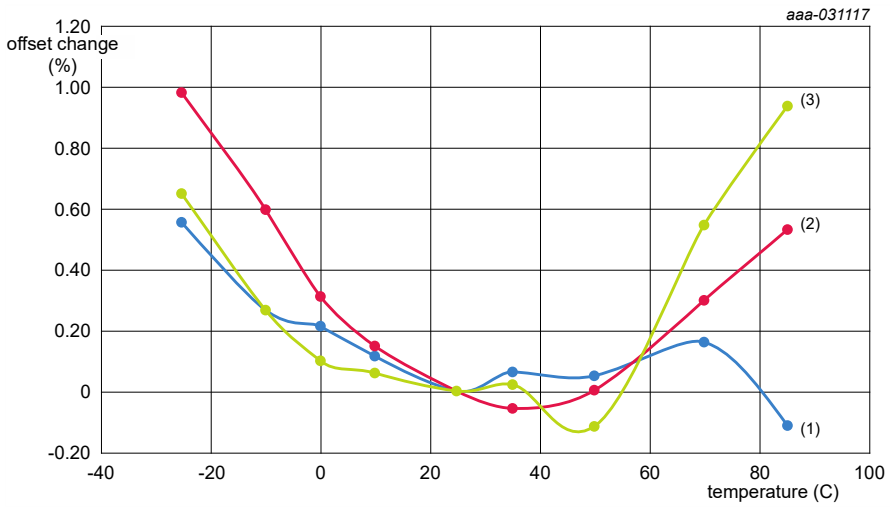
Table 74. ENOB for different input ranges and corrections...*continued*

Input range	Offset/gain corrected (2-point calibration)`	Quadratic corrected (3-point calibration)
50 nA	10	10.3
100 nA	9	10
500 nA	10	10.7
2.5 µA	9	9.7
5 µA	8.4	10.7
25 µA	8	9.3
250 µA	8	11.3

8.5 Impact of temperature on offset and gain error

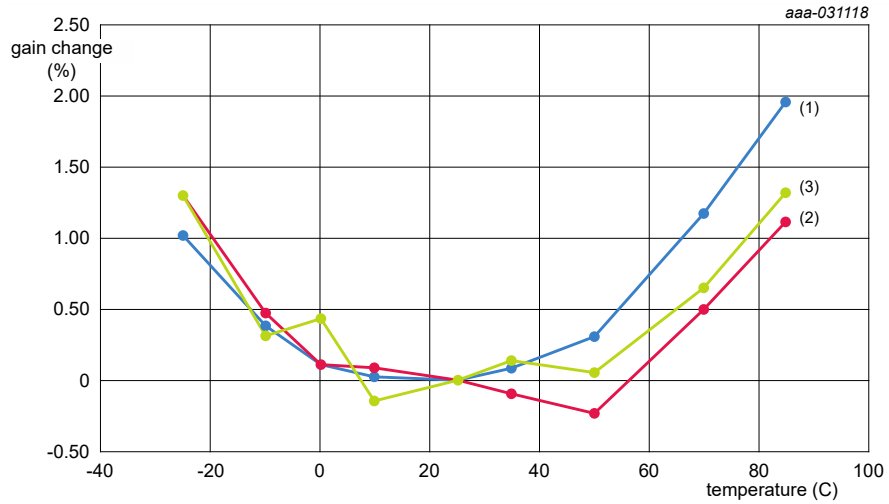
The current-to-digital converter is slightly sensitive to temperature variations, which becomes apparent when using the low range input settings (< 50 nA). Higher temperatures cause exponentially increasing leakage currents in the analog switching matrix and converter circuit, which result in an important offset and gain error above 50 °C.

The following graphs show the relative impact of temperature on the converted values. For example, operating the converter in the 250 µA range at 50 °C would result in an offset of −0.1 % (0.25 µA) and a gain of +0.06 %.



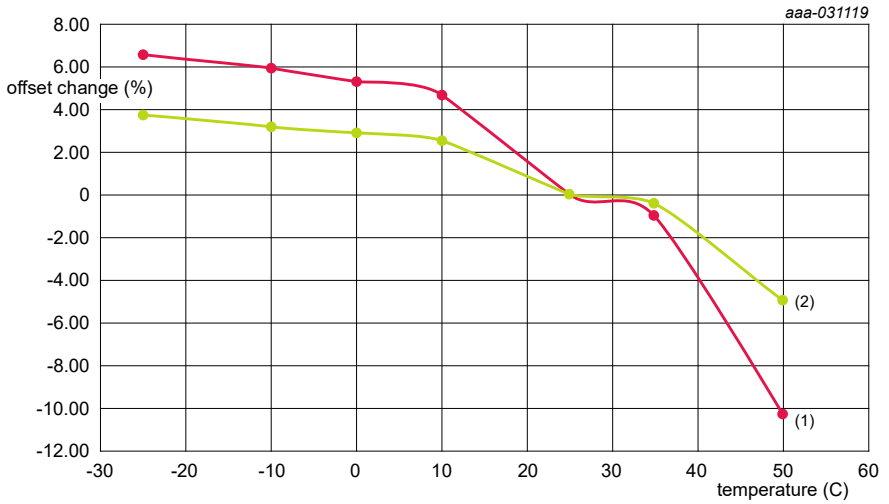
(1) 2.5 µA
(2) 25 µA
(3) 250 µA

Figure 14. Typical offset deviation due to temperature changes, relative to the offset at 25 °C and the selected range



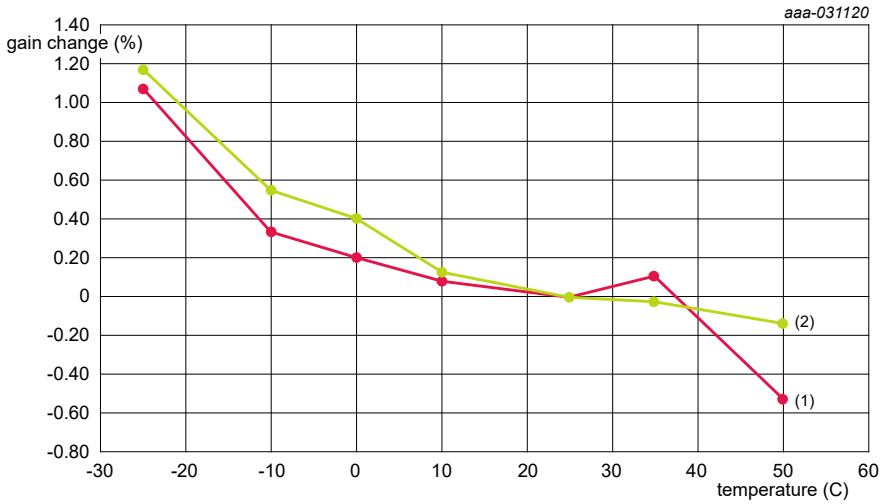
(1) 2.5 µA
(2) 25 µA
(3) 250 µA

Figure 15. Typical gain deviation due to temperature changes, relative to the gain at 25 °C and the selected range



(1) 25 µA
(2) 50 µA

Figure 16. Typical offset deviation due to temperature changes, relative to the offset at 25 °C and the selected range



(1) 25 µA

(2) 50 µA

Figure 17. Typical gain deviation due to temperature changes, relative to the gain at 25 °C and the selected range

8.6 Register descriptions

The register addresses for the Current-to-Digital converter are listed in Table 75. The register base address is 0x4006 8000.

The reset values reflect the data stored in used bits only, it does not include the content of reserved bits.

Table 75. Register overview: Current-to-Digital converter (base address 0x4006 8000)

Name	Access	Address offset	Description	Reset value [1]
CR	R/W	0x000	control register	0x0000 0000
DR	R	0x004	data register. Contains result of conversion	0x0000 0000
SR	R	0x008	status register	0x0000 0000
SP0	R/W	0x00C	setup register 0: set ADC mode settings	0x0000 0043
SP1	R/W	0x010	setup register 1: integration time settings	0x0000 0000
SP2	R/W	0x014	setup register 2: integration time calibration	0x0000 0009
SP3	R/W	0x018	setup register 3: input gain settings	0x0000 0000
MUXA	R/W	0x01C	input multiplexer settings	0x0000 0000
TLO	R/W	0x020	low value threshold register	0x0000 0000
THI	R/W	0x024	high value threshold register	0x0000 FFFF
IMSC	R/W	0x028	interrupt Mask Set and Clear Register	0x0000 0000
RIS	R	0x02C	raw Interrupt Status Register	0x0000 0000
MIS	R	0x030	masked Interrupt Status Register	0x0000 0000
ICR	W	0x034	interrupt Clear Register	n/a

[1] Reset value reflects the data stored in used bits only. It does not include content of reserved bits.

8.6.1 Current-to-digital control register (CR)

The current-to-digital converter control register sets the operation mode and start/stops the conversions.

Table 76. CR register (address 0x4006 8000) bit description

Bit	Symbol	Description	Reset value	Access
0	START	start bit. I2D clears this bit at the end of the measurement.	0	R/W
1	STOP	stop bit. This bit stops the converter after the ongoing conversion in Continuous mode. There is no effect in single-shot mode.	0	W
2	MODE	Conversion mode: (0) single-shot, (1) continuous.	0	R/W
31:3	-	reserved	0	-

8.6.2 Current-to-digital data register (DR)

The DR register holds a value proportional to the measured current as determined by the Current-to-Digital converter at the given input scaling. Reading the DR register clears the Current-to-Digital interrupts.

Table 77. DR register (address 0x4006 8004) bit description

Bit	Symbol	Description	Reset value	Access
15:0	DATA	result of last conversion, unsigned 16-bit integer	0	R
16	OFW	(0) input in range, (1) overflow	0	R
31:17	-	reserved	0	-

8.6.3 Current-to-digital status register (SR)

The SR register holds the status of the Current-to-Digital converter.

Table 78. SR register (address 0x4006 8008) bit description

Bit	Symbol	Description	Reset value	Access
0	OFW	input out of range (overflow)[1]	0	R
1	UFW	input out of range (underflow)[2]	0	R
31:2	-	reserved	0	-

[1] Range too low indicates that the converted digital value is too high for the configured range. The 16-bit DATA value in the DR register is inaccurate.
[2] Range too high indicates that the converted digital value is too low for the configured range. The 16-bit DATA value in the DR register is inaccurate.

8.6.4 Current-to-digital setup register 0 (SP0)

The SP0 register controls the resolution and mode settings for the ADC of the Current-to-Digital converter.

Table 79. SP0 register (address 0x4006 800C) bit description

Bit	Symbol	Description		Reset value	Access
2:0	INT	integration time setting		0	R/W
		000b	reserved		
		001b	16 ms		
		010b	20 ms		
		011b	100 ms		
		100b	200 ms		
		101b	400 ms		
		110b	reserved		
		111b	custom integration time as defined by the SP1 register		
5:3	-	reserved		0	-
6	GAIN0	ADC internal gain setting		1	R/W
		0	high gain		
		1	low gain		
7	-	reserved		0	R/W

Table 79. SP0 register (address 0x4006 800C) bit description...*continued*

Bit	Symbol	Description	Reset value	Access
8	SELIOFF	enable offset current	0	R/W
9	IOFF0	select offset current 0	0	R/W
10	IOFF1	select offset current 1	0	R/W
31:11	-	reserved	0	-

8.6.5 Current-to-digital setup register 1 (SP1)

The SP1 register contains the custom integration time.

Table 80. SP1 register (address 0x4006 8010) bit description

Bit	Symbol	Description	Reset value	Access
15:0	CINT	custom integration time in ms	0	R/W
31:16	-	reserved	0	-

8.6.6 Current-to-digital setup register 2 (SP2)

The SP2 register contains calibration setting for the integration time (system clock to time calibration factor). When a custom integration time is selected in register SP0, this register has to be reset to 999 (decimal).

Table 81. SP2 register (address 0x4006 8014) bit description

Bit	Symbol	Description	Reset value	Access
13:0	CAL	integration time calibration to clock × clock pulse to count 1 ms	9h	R/W
31:14	-	reserved	0	-

8.6.7 Current-to-digital setup register 3 (SP3)

The SP3 register contains the settings of the input current scaler.

Table 82. SP3 register (address 0x4006 8018) bit description

Bit	Symbol	Description		Reset value	Access
0	GAIN_EN	input gain amplifier enable / bypass		1	R/W
		0	gain amplifier bypassed		
		1	gain amplifier enabled		
3:1	IGAIN [1]	input gain / attenuation settings		0	R/W
		000b	no gain (analog input 1:1 output to I/F)		
		001b	multiply input current by 2 (1:2)		
		010b	multiply input current by 10 (1:10)		
		011b	divide input current by 2 (2:1)		
		100b	divide input current by 10 (10:1)		
		101b	divide input current by 100 (100:1)		
		110b	reserved		
		111b	reserved		
31:4		reserved		0	-

[1] Not all combinations of IGAIN and GAIN0 are possible

8.6.8 Current-to-digital input multiplexer (MUX)

The MUX register contains the settings for the multiplexer of the Current-to-Digital converter.

Table 83. MUX register (address 0x4006 801C) bit description

Bit	Symbol	Description	Reset value	Access
5:0	MUX	each bit represents an input of the switching matrix. Logic 1: connected to input, logic 0: disconnected	0	R/W
31:6	-	reserved	0	-

8.6.9 Current-to-digital low-value threshold register (TLO)

The TLO register contains the low threshold value. If the measured bit value is lower than this threshold value, and if corresponding bit in the IMSC register is set, then an interrupt is generated.

Table 84. TLO register (address 0x4006 4020) bit description

Bit	Symbol	Description	Reset value	Access
15:0	TLO	low threshold value	0	R/W
31:16	-	reserved	0	-

8.6.10 Current-to-digital high-value threshold register (THI)

The THI register contains the high threshold value. If the measured bit value is higher than this threshold value, and if corresponding bit in the IMSC register is set, then an interrupt is generated.

Table 85. THI register (address 0x4006 4024) bit description

Bit	Symbol	Description	Reset value	Access
15:0	THI	high threshold value.	0xFFFF	R/W
31:16	-	reserved	0	-

8.6.11 Current-to-digital interrupt mask Set/Clear register (IMSC)

This register controls whether each of the three possible interrupt conditions in the Current-to-Digital converter are enabled.

Table 86. IMSC register (address 0x4006 8028) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	software should set this bit to enable interrupt when a conversion value is ready in DR	0	R/W
1	LO	software should set this bit to enable interrupt when the value of the conversion is lower than the value stored in TLO	0	R/W
2	HI	software should set this bit to enable interrupt when the conversion is higher than the value stored in THI	0	R/W
31:3	-	reserved	0	-

8.6.12 Current-to-digital raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless of whether the interrupt is enabled in the IMSC register.

Table 87. RIS register (address 0x4006 802C) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	this bit is logic 1 when an I2D conversion has finished	0	R
1	LO	this bit is logic 1 when an I2D conversion has finished and the value is below the threshold value set in the TLO register	0	R
2	HI	this bit is logic 1 when an I2D conversion has finished and the value exceeds the threshold value set in the THI register	0	R
31:3	-	reserved	0	-

8.6.13 Current-to-digital masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the IMSC registers. When an I2D converter interrupt occurs, the interrupt service routine should read this register to determine the cause or causes of the interrupt.

Table 88. MIS register (address 0x4006 8030) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	this bit is logic 1 when an I2D conversion has finished and this interrupt is enabled	0	R
1	LO	this bit is logic 1 when an I2D conversion has finished, the value is below the TLO threshold value and this interrupt is enabled	0	R
2	HI	this bit is logic 1 when an I2D conversion has finished, the value exceeds the THI threshold value and this interrupt is enabled	0	R
31:3	-	reserved	0	-

8.6.14 Current-to-digital interrupt clear register (ICR)

Software can write one or more one(s) to this write-only register, to clear the corresponding interrupt condition or conditions in the Current-to-Digital converter.

Note: *The other two interrupt conditions can be cleared by reading the DR register or disabled by clearing the corresponding bit in IMSC registers.*

Table 89. ICR register (address 0x4006 4034) bit description

Bit	Symbol	Description	Reset value	Access
0	RDYI	writing a logic 1 clears the conversion ready interrupt	0	W
1	LO	writing a logic 1 clears the low threshold interrupt	0	W
2	HI	writing a logic 1 clears the high threshold interrupt	0	W
31:3	-	reserved	0	-

9 Analog-to-digital/digital-to-analog converter (ADCDAC0)

9.1 About this chapter

This chapter describes the setup and use of the Analog-to-Digital / Digital-to-Analog converter.

9.2 Features

- 12-bit ADC operation at 80 kSamples/s.
- 12-bit DAC operation with hold amplifier.

9.3 General description

The ADCDAC0 peripheral is based on a 12-bit successive-approximation charge-redistribution analog-to-digital converter. The peripheral acts as either an analog-to-digital converter or a digital-to-analog converter, depending on which start bit is written to the control register CR. Requests are handled in round-robin fashion.

Due to the hold amplifier, the DAC output remains constant while the block is used as an ADC.

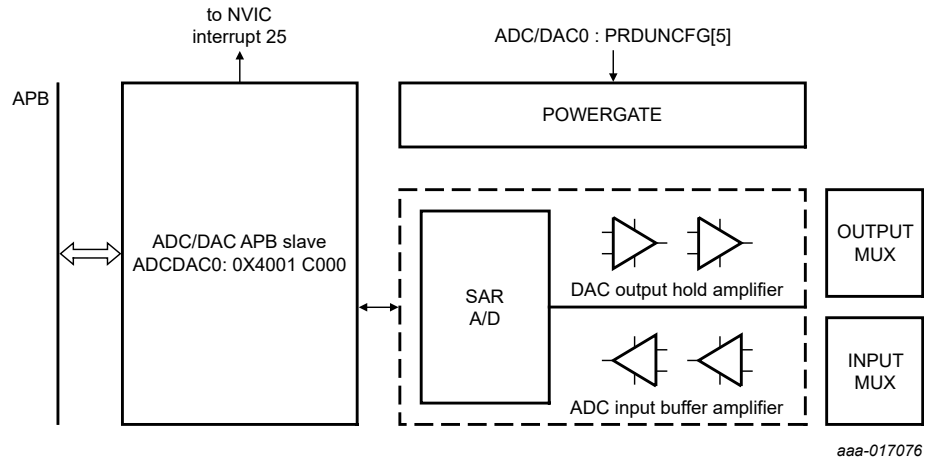


Figure 18. Analog-to-Digital / Digital-to-Analog block diagram

9.4 Input multiplexer and output switch matrix

The ADC input and DAC output are connected to the analog buses via an analog multiplexer. Table 90 shows the connections of the multiplexer.

Table 90. Connections to ADC input and output multiplexers

AMUX input	Source / target	Description
0	ana_extbus0	(external) pin ANA0_0
1	ana_extbus1	(external) pin ANA0_1
2	ana_extbus2	(external) pin ANA0_2
3	ana_extbus3	(external) pin ANA0_3
4	ana_extbus4	(external) pin ANA0_4

Table 90. Connections to ADC input and output multiplexers...*continued*

AMUX input	Source / target	Description
5	ana_extbus5	(external) pin ANA0_5
6	reserved	reserved
7	reserved	reserved
8	reserved	reserved
9	reserved	reserved
10	reserved	reserved
11	reserved	reserved

9.5 Register descriptions

The register addresses for the Analog-to-Digital / Digital-to-Analog converter are listed in Table 103.

Table 91. Register overview: Analog-to-Digital / Digital-to-Analog converter (base address 0x4001 C000)

Name	Access	Address offset	Description	Reset value[1]
CR	R/W	0x000	control register	0x0000 0000
ADCDR	R	0x004	ADC Data register. Contains result of Analog-to-Digital conversion	0x0000 0000
DACDR	W	0x008	DAC Data register. Contains value to be converted to analog voltage	0x0000 0000
SP0	R/W	0x010	ADCDAC setup register	0x0000 0000
ADCMUX	R/W	0x018	ADC input multiplexer setting	0x0000 0000
DACMUX	R/W	0x01C	DAC output switch setting	0x0000 0000
ADCTLO	R/W	0x020	low value threshold register	0x0000 0000
ADCTHI	R/W	0x024	high value threshold register	0x0000 0000
IMSC	R/W	0x028	interrupt Mask Set and Clear register	0x0000 0000
RIS	R	0x02C	raw Interrupt Status register	0x0000 0008
MIS	R	0x030	masked Interrupt Status register	0x0000 0000
ICR	W	0x034	interrupt Clear register	n/a

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

9.5.1 ADC control register (CR)

The ADCDAC control register controls the operation of the SAR ADC/DAC.

Table 92. CR register (address 0x4001 C000) bit description

Bit	Symbol	Description	Reset value	Access
0	ADCSTART	start bit for Analog-to-Digital conversion. ADCDAC clears this bit at the end of the measurement	0	R/W
1	DACSTART	start bit for Digital-to-Analog conversion. When the analog level is available on the output bus, the ADCDAC clears this bit	0	R/W
2	ADCCONT	set ADC to Continuous mode	0	R/W
3	DACCONT	set DAC to Continuous mode	0	R/W
4	ADCSTOP	stop ADC conversion after current conversion (in Continuous mode)	0	R/W
5	DACSTOP	stop DAC conversion after current conversion (in Continuous mode)	0	R/W
31:6	-	reserved	-	-

9.5.2 ADC data register (ADCDR)

The ADCDR register holds a value proportional to the measured voltage as determined by the Analog-to-Digital converter.

Table 93. ADCDR register (address 0x4001 C004) bit description

Bit	Symbol	Description	Reset value	Access
11:0	ADCDATA	result of last conversion, unsigned 12-bit integer	0x000	R
31:12	-	reserved	-	-

9.5.3 DAC data register (DACDR)

The DACDR register holds the value to be converted to a voltage. This conversion will occur at the next write of the DACSTART bit in the CR register.

Table 94. DACDR register (address 0x4001 C008) bit description

Bit	Symbol	Description	Reset value	Access
11:0	DACDATA	value to be converted to voltage, unsigned 12-bit integer	0x000	R
31:12	-	reserved	-	-

9.5.4 ADC status register (SR)

The ADCSR register holds the status of the ADCDAC.

Table 95. SR register (address 0x4001 C00C) bit description

Bit	Symbol	Description	Reset value	Access
31:0	-	reserved	-	-

9.5.5 ADCDAC setup register (ADCSP)

The ADCSP register holds the setup of the ADC.

Table 96. ADCSP register (address 0x4001 C010) bit description

Bit	Symbol	Description		Reset value	Access
0	ADOFFSETCAL	measurement reference voltage. Setting should correspond to ADC input range setting.		1	R/W
		0	0.5 V		
		1	0.9 V		
1	ADWIDERANGE	input range		1	R/W
		0	0 V - 1 V		
		1	0 V - 1.6 V		
2	BOOSTSUPREF	boost the drive capability of the reference/supply buffers		0	R/W
3	CURRBACKUP	enable additional bias current within the reference block		0	R/W
4	LESSSLEW	boost current and therefore the speed of DAC output		0	R/W
5	SLOWCLK	reduce overall speed of ADCDAC		0	R/W
6	DYNAMICSLP	reduce the power of input and output amp if they are not used immediately		0	R/W
7	BOOSTINAMP	boost current and therefore speed of ADC input amplifier		0	R/W
8	BOOSTOUTAMP	boost current and therefore speed of DAC output amplifier		0	R/W
9	ONEACTIVE	disable the non-selected function (ADC or DAC)		0	R/W
31:10	-	reserved		-	-

9.5.6 ADC input multiplexer register (ADCMUX)

ADCMUX holds the settings for the input multiplexers.

Table 97. ADCMUX register (address 0x4001 C018) bit description

Bit	Symbol	Description		Reset value	Access
19:0	ADCMUX	each bit represents an input of the switching matrix		0	R/W
		0	disconnected		
		1	connected to input		
31:20	-	reserved		-	-

9.5.7 DAC output switch register (DACMUX)

DACSW holds the settings for the output switch matrix.

Table 98. DACMUX register (address 0x4001 C01C) bit description

Bit	Symbol	Description		Reset value	Access
19:0	DACSW	each bit represents an input of the switching matrix		0	R/W
		0	disconnected		
		1	connected to input		
31:20	-	reserved		-	-

9.5.8 ADCDAC low-value threshold register (ADCTLO)

ADCTLO contains the low threshold value. The measured bit value is compared to this value. If the measured value is lower, and the corresponding bit in the ADCIMSC register is set, an interrupt is generated.

Table 99. ADCTLO register (address 0x4001 C020) bit description

Bit	Symbol	Description	Reset value	Access
15:0	ADCTLO	low threshold value	0x0000	R/W
31:16	-	reserved	-	-

9.5.9 ADCDAC high-value threshold register (ADCTHI)

ADCTHI contains the low threshold value. The measured bit value is compared to this value. If the measured value is higher, and the corresponding bit in the ADCIMSC register is set, an interrupt is generated.

Table 100. ADC0THI register (address 0x4001 C024) bit description

Bit	Symbol	Description	Reset value	Access
15:0	ADCTHI	high threshold value	0x0000	R/W
31:16	-	reserved	-	-

9.5.10 ADCDAC interrupt mask set/clear register (IMSC)

This register controls whether each of the three possible interrupt conditions in the ADC are enabled.

Table 101. IMSC register (address 0x4001 C028) bit description

Bit	Symbol	Description	Reset value	Access
0	DACRDYI	software should set this bit to enable interrupt when the DAC finished a conversion	0	R/W
1	ADCRDYI	software should set this bit to enable interrupt when an ADC conversion value is ready in ADCDR	0	R/W
2	ADCLO	software should set this bit to enable interrupt when the value of the conversion is lower than the value stored in ADCTLO	0	R/W
3	ADCHI	software should set this bit to enable interrupt when the value of the conversion is higher than the value stored in ADCTHI	0	R/W

Table 101. IMSC register (address 0x4001 C028) bit description*...continued*

Bit	Symbol	Description	Reset value	Access
31:4	-	reserved	-	-

9.5.11 ADCDAC raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless of whether the interrupt is enabled in the IMSC register.

Table 102. RIS register (address 0x4001 C02C) bit description

Bit	Symbol	Description	Reset value	Access
0	DACRDYI	this bit is logic 1 when a D/A conversion has finished	0	R
1	ADCRDYI	this bit is logic 1 when an A/D conversion has finished	0	R
2	ADCLOI	this bit is logic 1 when an A/D conversion has finished and the value is below the threshold value set in the ADCTLO register	0	R
3	ADCHII	this bit is logic 1 when an A/D conversion has finished and the value exceeds the threshold value set in the ADCTHI register	0	R
31:4	-	reserved	-	-

9.5.12 ADCDAC masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the IMSC register. When an ADCDAC interrupt occurs, the interrupt service routine should read this register to determine the cause or causes of the interrupt.

Table 103. MIS register (address 0x4001 C030) bit description

Bit	Symbol	Description	Reset value	Access
0	DACRDYI	this bit is logic 1 when a D/A conversion has finished and this interrupt is enabled	0	R
1	ADCRDYI	this bit is logic 1 when an A/D conversion has finished and this interrupt is enabled	0	R
2	ADCLOI	this bit is logic 1 when an A/D conversion has finished, the value is below the ADCTLO threshold value and this interrupt is enabled	0	R
3	ADCHII	this bit is logic 1 when an A/D conversion has finished, the value exceeds the ADCTHI threshold value and this interrupt is enabled	0	R
31:4	-	reserved	-	-

9.5.13 ADCDAC interrupt clear register (ICR)

Software can write one or more logic 1s to this write-only register, to clear the corresponding interrupt condition or conditions in the Capacitance-to-Digital converter.

Note: *The other two interrupt conditions can be cleared by reading the ADCDR register or disabled by clearing the corresponding bit in the IMSC register.*

Table 104. ADC0ICR register (address 0x4001 C034) bit description

Bit	Symbol	Description	Reset value	Access
0	DACRDYIC	writing a logic 1 clears the D/A conversion ready interrupt	0	W
1	ADCRDYC	writing a logic 1 clears the A/D conversion ready interrupt	0	W
2	ADCLOC	writing a logic 1 clears the low threshold interrupt	0	W
3	ADCHIC	writing a logic 1 clears the high threshold interrupt	0	W
31:4	-	reserved	-	-

10 Analog signal buses

10.1 About this chapter

This chapter describes the analog signal buses.

10.2 General description

The NHS31xx family has two sources of analog signals: external (via pads) and internal (from the DAC). The ‘ana_ext bus is connected through a 3.3 V selector to the analog pins. The IOCON_ANA0_x registers in the IOCON block (see Section 23.3.13 to Section 23.3.19) control it. Figure 19 schematically shows the bus structure.

Each analog signal bus can be grounded using the corresponding bit in the ANABUSGROUND register of IOCON.

To use an analog bus, the user must first disconnect the bus from VSS using the appropriate bit in the ANABUSGROUND register of the IOCON block. If the analog bus has to receive an external signal, the FUNC bit in the corresponding ANA0_x register must be set.

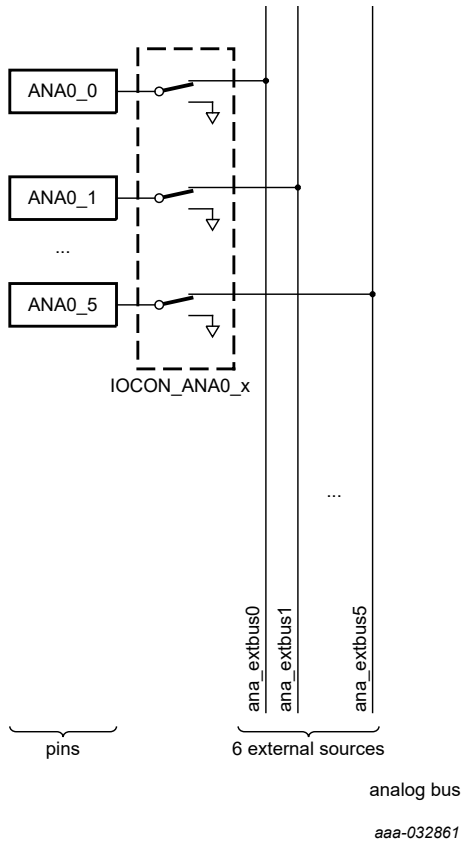


Figure 19. NHS31xx analog bus structure

The pads are connected to analog bus lines. The different converters connect through their own analog switch matrix to the bus lines.

11 RFID/NFC communication unit

11.1 About this chapter

This chapter describes the use of the RFID/NFC Communication unit.

11.2 Introduction

Near field communication (NFC) is a technology that enables smartphones, and many other devices, to establish radio communication by bringing them into close proximity. Typically NFC has a range of 10 cm (3.9 inch) or less.

11.3 Features

- ISO/IEC 14443 A 1-3 compatible
- MIFARE (Ultralight) compatible
- NFC Forum Type 2 compatible
- Easy interfacing with standard user memory space READ/WRITE commands
- Passive operation possible
- 50 pF internal capacitance for compatibility with standard NFC antennas

11.4 General description

The RFID/NFC interface allows communication using 13.56 MHz proximity signaling. It is based on the MIFARE Ultralight EV1 PICC. The RFID interface works internally at 6.78 MHz. This clock is recovered from the RF field and is independent of the NHS31xx system clock.

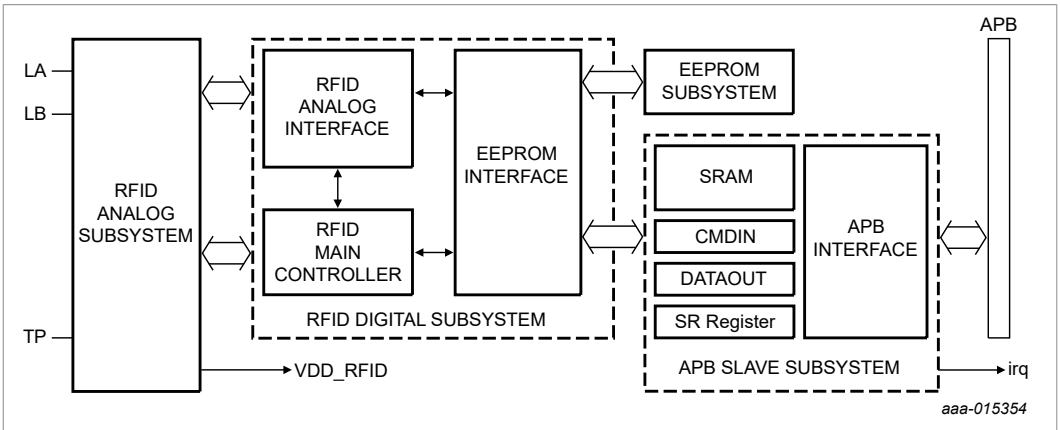


Figure 20. Block diagram of the RFID/NFC interface

The CMDIN, CMDOUT, SR and SRAM are mapped in the user memory space of the RFID core. The RFID READ and WRITE commands allows wireless communication via this shared memory. By only allowing access after password authentication by the MIFARE protocol, the READ/WRITE operation can be blocked.

The firmware running on the ARM Cortex-M0+ core interprets commands written to the CMDIN and CMDOUT registers. Typically, the reader would first gain access by SELECT and AUTH (if needed).

11.5 Pin description

The RFID communication module interfaces with the LA and LB external pins.

There is a 50 pF internal capacitance between the pins.

More background information can be found in Ref. 2.

11.6 Functional description

Functionality and design of the MIFARE Ultralight EV1 is described in Ref. 1. In this document, we only focus on the broad operation and interfacing with the APB.

11.6.1 ISO 14443 A and MIFARE operation

Figure 21 shows the memory map as seen from RF interface. Segment 0 is 0x00 to 0x03. The user can set the boundary of Segment 1 to Segment 2.

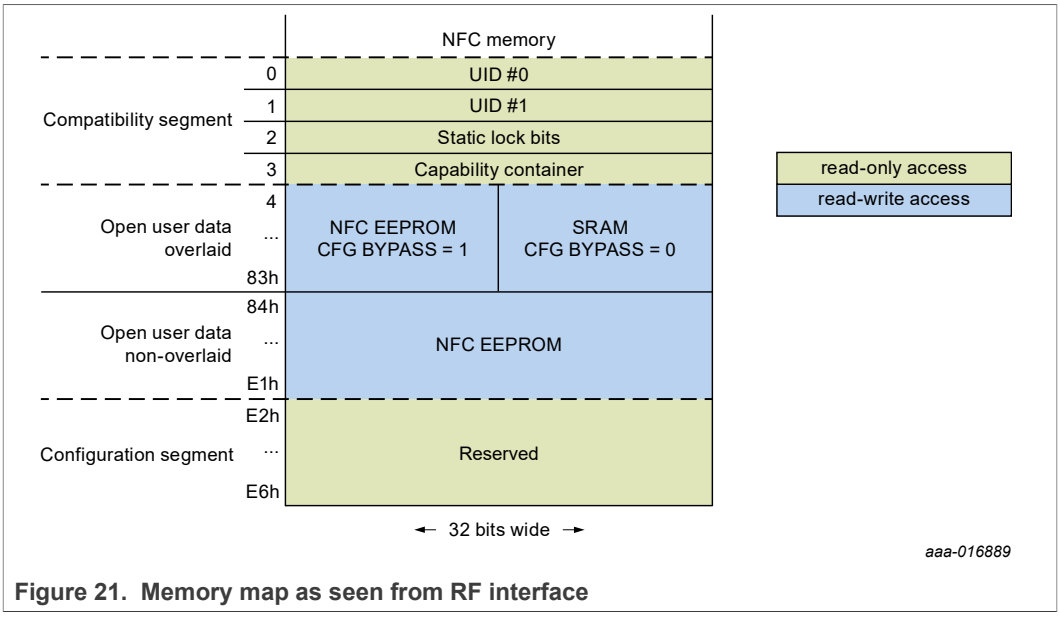


Figure 21. Memory map as seen from RF interface

231 pages are accessible from the RF interface. Each page is 4 bytes wide.

- Pages 0 up to and including 3 are read-only.
- Pages 4 up to and including 131 are R/W. The ARM processor can also read these pages and write to them, provided the BYPASS field in the CFG register is cleared.
- Pages 131 up to and including 225 are R/W. The ARM processor cannot access these pages.
- Pages 226 up to and including 230 are read-only.

On the MIFARE ultralight commands:

- The READ command always returns 4 pages of 4 bytes to the reader. When reading at the end of the buffer, the returned bytes are padded with 0x00.
- The FAST_READ command accepts ranges from 4 to 0xE6.
- The WRITE command writes one page.
- The COMP_WRITE command sends 16 data bytes to the NHS31xx. However, only the least significant 4 bytes are written to the addressed page.

11.6.2 NHS31xx family communication protocol

Communication between the RFID/NFC reader and the NHS31xx is initiated once the reader selects NHS31xx via ISO14443 commands. Access to the shared memory space is arbitrated between the RF side and the APB side by the arbitration unit. RFID-EEPROM access is only possible from RFID the side, and only for addresses not masked by the shared memory or the shared registers.

The reader can execute READ or WRITE commands on the shared memory or on the registers. Interpretation of the communication / messages written to the SRAM is done by firmware.

11.6.2.1 Device-specific GET_VERSION command

The `GET_VERSION` command is used to retrieve information on tags compatible with the MIFARE family. It provides product version, storage size and product info required to identify the product. This command is available on all MIFARE-compatible products and provides a common way of identifying products across platforms and evolution steps.

The `GET_VERSION` command has no arguments and replies the version information for the specific type. The response structure with the NHS-specific values is shown in the table below:

Table 105. GET_VERSION response for NHS devices

Byte	Description	Value	Interpretation
0	Fixed header	00h	
1	Vendor ID	04h	NXP Semiconductors
2	Product type	04h	NTAG
3	Product subtype	06h	NHS
4	Major product version	00h	
5	Minor product version	00h	
6	Size	13h	Memory size is between 2^9 and 2^{10}.
7	Protocol type	03h	ISO/IEC 14443-3 compliant

All NHS devices have a user memory of 888 bytes (222 pages). Of these, 512 bytes (128 pages) are accessible (R/W) by the ARM. The first 4 pages (R) and the last 90 pages (R/W) are only accessible by the PCD.

11.7 Register descriptions

Data is stored in a communication buffer of 4 kbit (128 × 4 bytes), while commands are read or written from/to the CMDIN and CMDOUT registers.

The reset values reflect the data stored in used bits only, it does not include the content of reserved bits. The addresses of the SR, CMDIN, CMDOUT, and buffer memory are different when seen from APB (Table 106) or RFID side (Table 107).

Table 106. RFID/NFC communication block register and buffer memory space overview: APB side

Name	Access	Address offset	Description	Reset value
CFG	R/W	0x000	RFID/NFC configuration register	0x0000 0000

Table 106. RFID/NFC communication block register and buffer memory space overview: APB side*...continued*

Name	Access	Address offset	Description	Reset value
SR	R	0x004	RFID/NFC status register	0x0000 0000
CMDIN	R	0x008	RFID/NFC incoming command	0x0000 0000
CMDOUT	R/W	0x00C	RFID/NFC outgoing command	0x0000 0000
TARGET	R/W	0x010	RFID/NFC target page address register	0x0000 0004
LAST_ACCESS	R	0x014	raw Interrupt status register	0x0000 0000
IMIS	R/W	0x018	interrupt mask register	0x0000 0000
RIS	R	0x01C	raw Interrupt status register	0x0000 0000
MIS	R	0x020	masked interrupt status register	0x0000 0000
IC	W	0x024	interrupt clear register	0x0000 0000
SHARED_MEM	R/W	0x100 - 0x2FC	buffer memory space (128 words)	undefined

Table 107. RFID/NFC communication block special registers and buffer memory space overview: RFID side

Name	Access	Address offset	Description	Reset value
SHARED_MEM	R/W	0x04 - 0x83	buffer memory space (128 words)	0
SR	R	0x84	RFID/NFC status register	0x0000 0000
CMDIN	R/W	0x85	RFID/NFC incoming command	0x0000 0000
CMDOUT	R	0x86	RFID/NFC outgoing data	0x0000 0000

11.7.1 RFID/NFC configuration register (CFG)

This register contains the configuration data for the RFID block.

Table 108. CFG register (address 0x4005 8000) bit description

Bit	Symbol	Description	Reset value	Access
0	BYPASS	when set, the shared memory interface and registers are deactivated, and the external reader gets full access to the RFID/NFC EEPROM.	0	R/W
31:1	-	reserved	0	-

11.7.2 RFID/NFC status register (SR)

This register contains the status of the RFID block.

Table 109. SR register (APB address 0x4005 8004, RF address 0x84) bit description

Bit	Symbol	Description	Reset value	Access[1]
0	POR	power-on detected in the RFID analog core	0	R
1	1V2	rectifier < 1.2 V output warning	0	R
2	1V5	rectifier < 1.5 V output warning	0	R
3	PLL	RFID PLL locked	0	R
4	SEL	RFID selected	0	R
5	AUTH	RFID password for access authenticated	0	R
6	BYPASS	RFID interface in Bypass mode	0	R
7	-	reserved	0	R
15:8	RFID_CMD_	opcode of last RFID command received:	0	R
		0000 0000: REQA		
		0000 0001: WUPA		
		0000 0010: ANTICOLLISION		
		0000 0011: SELECT		
		0000 0100: HLTA		
		0000 0101: READ		
		0000 0110: COMP_WRITE		
		0000 0111: WRITE		
		0000 1000: INCR_CNT		
		0000 1001: READ_CNT		
		0000 1010: PWD_AUTH		
		0000 1011: GET_VERSION		
		0000 1100: FAST_READ		
		0000 1101: READ_SIG		
		0000 1111: CHECK_TEARING_EVENT		
31:12	-	reserved	-	-

[1] Read access from the RFID side, Write access from the APB side.

11.7.3 RFID/NFC incoming command register (CMDIN)

This register contains an NHS3-specific incoming command. Reading this register by the ARM clears the corresponding interrupt. This register is writable through the RFID wireless interface.

Table 110. CMDIN register (APB address 0x4005 800C, RF address 0x85) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CMDIN	incoming command	0x0000 0000	R

11.7.4 RFID/NFC outgoing data register (CMDOUT)

This register contains NHS3-specific outgoing data. This register can be read over the RFID wireless interface.

Table 111. CMDOUT register (APB address 0x4005 8010, RF address 0x86) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CMDOUT	outgoing data	0x0000 0000	R/W

11.7.5 RFID/NFC target address register (TARGET)

This register contains an NHS3-specific outgoing command. This register can be read over the RFID wireless interface.

Table 112. TARGET register (address 0x4005 8010) bit description

Bit	Symbol	Description	Reset value	Access
7:0	PAGE	target (RF) page address	0x04	R/W
31:8	-	reserved	0	-

11.7.6 RFID/NFC last accessed address register (LAST_ACCESS)

This register contains the address of the last accessed pages. This address is an address as seen from the RF side. Both start and end of read/write address range are shown.

Table 113. LAST_ACCESS register (address 0x4005 8014) bit description

Bit	Symbol	Description	Reset value	Access
7:0	LAST_ACCESS_END	last accessed (RF) page address (end of range)	0	R
15:8	LAST_ACCESS_START	last accessed (RF) page address (start of range)	0	R
16	DIR	direction of last accessed page (0) READ, (1) WRITE	0	R
31:17	-	reserved	0	-

11.7.7 RFID/NFC interrupt mask register (IMSC)

This register controls whether RFID read/write access interrupt condition is enabled.

Table 114. IMSC register (address 0x4005 8014) bit description

Bit	Symbol	Description	Reset value	Access
0	RFPOWER	set to enable interrupt when RFID power is detected	0	R/W
1	RFSELECT	set to enable interrupt when reader selects tag	0	R/W
2	MEMREAD	set to enable interrupt when reader reads from shared memory (SHARED_MEM)	0	R/W
3	MEMWRITE	set to enable interrupt when reader writes to shared memory (SHARED_MEM)	0	R/W

Table 114. IMSC register (address 0x4005 8014) bit description*...continued*

Bit	Symbol	Description	Reset value	Access
4	CMDWRITE	set to enable interrupt when reader writers to CMDIN register	0	R/W
5	CMDREAD	set to enable interrupt when reader reads the CMDOUT register	0	R/W
6	TARGETWRITE	set to enable interrupt when reader writes to address specified in the TARGET register	0	R/W
7	TARGETREAD	set to enable interrupt when reader reads from address specified in the TARGET register	0	R/W
8	NFCOFF	set to enable interrupt when external reader powers down RFID/NFC front-end	0	R/W
31:9	-	reserved	-	-

11.7.8 RFID/NFC raw interrupt status register (RIS)

This read-only register contains a 1 for each interrupt condition that is asserted, regardless of whether the interrupt is enabled in the IMSC register.

Table 115. RIS register (address 0x4005 801C) bit description

Bit	Symbol	Description	Reset value	Access
0	RFPOWER	RFID power is detected	0	R/W
1	RFSELECT	reader selected tag	0	R/W
2	MEMREAD	reader reads from shared memory (SHARED_MEM)	0	R/W
3	MEMWRITE	reader writes to shared memory (SHARED_MEM)	0	R/W
4	CMDWRITE	reader writers to CMDIN register	0	R/W
5	CMDREAD	reader reads the CMDOUT register	0	R/W
6	TARGETWRITE	reader writes to address specified in the TARGET register	0	R/W
7	TARGETREAD	reader reads from address specified in the TARGET register	0	R/W
8	NFCOFF	RFID/NFC front-end is not powered or clocked	0	R/W
31:9	-	reserved	0	-

11.7.9 RFID/NFC interrupt clear register (ICR)

Software can write one or more one(s) to this write-only register to clear the corresponding interrupt condition or conditions in the RFID/NFC controller.

Note: *The second interrupt conditions can be cleared by reading the CMDIN register.*

Table 116. ICR register (address 0x4005 8024) bit description

Bit	Symbol	Description	Reset value	Access
0	RFPOWER	write logic 1 to clear RFPOWER interrupt	0	W
1	RFSELECT	write logic 1 to clear RFSELECT interrupt	0	W
2	MEMREAD	write logic 1 to clear MEMREAD interrupt	0	W
3	MEMWRITE	write logic 1 to clear MEMWRITE interrupt	0	W
4	CMDWRITE	write logic 1 to clear CMDWRITE interrupt	0	W
5	CMDREAD	write logic 1 to clear CMDREAD interrupt	0	W

Table 116. ICR register (address 0x4005 8024) bit description*...continued*

Bit	Symbol	Description	Reset value	Access
6	TARGETWRITE	write logic 1 to clear TARGETWRITE interrupt	0	W
7	TARGETREAD	write logic 1 to clear TARGETREAD interrupt	0	W
8	NFCOFF	write logic 1 to clear NFCOFF interrupt	0	W
31:9	-	reserved	0	-

11.7.10 RFID/NFC buffer memory

The buffer memory ranges from APB address 0x4005 8100 to 0x4005 82FC or RF address 0x04 to 0x83. The purpose of the buffer memory is to serve as an interface for the exchange of NFC or NDEF messages. The buffer memory contains random data after reset. This memory is word-addressable only.

12 Real-time clock (RTC)

12.1 About this chapter

This chapter describes the usage of the Real Time Clock (RTC) timer.

The RTC timer is identical on all members of the NHS31xx family.

12.2 Features

The Real Time Clock (RTC) timer provides a countdown timer generating a wake-up signal when it expires. The timer runs on a low speed clock and and is always active unless the VDDBAT is switched off.

The RTC timer generates the RTC interrupt signal (wake-up interrupt 12), which can be enabled according to the settings in the RTCIMSC register. The timer counts in "ticks" which are derived from the TFRO clock pulses via the calibration value stored in the RTCCAL register.

12.3 General description

The RTC module consists of 3 parts:

- The RTC core module, implementing the RTC timer itself
  This module runs in the 'always on' VDD domain and is implemented in 3.3 V logic.
- The AMBA APB slave interface
  This module allows configuration of the RTC core via an APB bus. This module runs in the switched power domain.
- Level-shifters
  This module interconnects the core and the APB interface. It contains all level-shifters and isolation gates necessary to interconnect both modules

12.4 Register descriptions

The CR register controls the RTC timer.

The reset values reflect the data stored in used bits only, they do not include the content of reserved bits.

Table 117. Register overview: RTC timer (base address 0x4005 4000)

Name	Access	Address offset	Description	Reset value
CR	R/W	0x000	RTC control register	0x0000 0000
SR	R	0x004	RTC status register	0x0000 0002
CAL	R/W	0x008	RTC calibration value	0x0000 8000
SLEEPT	R/W	0x00C	RTC Sleep time	0x0000 0000
VAL	R	0x010	current RTC counter	0x0000 0000
IMIS	R/W	0x014	interrupt mask register	0x0000 0000
RIS	R	0x018	raw Interrupt status register	0x0000 0000
MIS	R	0x01C	masked interrupt status register.	0x0000 0000

Table 117. Register overview: RTC timer (base address 0x4005 4000)*...continued*

Name	Access	Address offset	Description	Reset value
ICR	W	0x020	interrupt clear register.	0x0000 0000
STAT	R	0x024	access status register.	0x0000 0000
TIME	R/W	0x030	RTC Time register.	0x0000 0000

12.4.1 RTC control register (CR)

The CR register controls the operation of the RTC Timer unit. If the unit is disabled, nothing happens.

Table 118. RTC timer control register (address 0x4005 4000) bit description

Bit	Symbol	Description	Reset value
0	EN	logic 1: Enable the RTC Timer, logic 0: disable	0
1	AUTOSTART	logic 1: Automatic timer start when value is written to SLEEPT register, logic 0: manual start	1
2	START	writing a logic 1 starts the countdown from the value stored in SLEEPT until logic 0 is reached. Write logic 0 to clear. This bit is also cleared when the interrupt is asserted or a new value is written into the SLEEPT register	0
31:3	-	reserved	0

12.4.2 RTC status register (SR)

The SR register shows the status of the RTC Timer unit. SR[1] is identical to PMU.TMRCLKCTRL[0].

For the RTC to count down time (RUN=1) the following conditions must be fulfilled:

1. it must be powered.
2. it must be receiving 32 kHz pulses.
3. it must be started.

Table 119. RTC Timer status register (address 0x4005 4004) bit description

Bit	Symbol	Description	Reset value
0	EN	logic 1: RTC timer enabled, logic 0: disabled	0
1	-	reserved	0
2	FREEZE	asserted near to the edges of the internal tick clock. Values written to the SLEEPT register during FREEZE=1 are delayed until after this condition clears	0
3	RUN	logic 1: RTC timer is running	0
31:4	-	reserved	0

12.4.3 RTC calibration register (CAL)

The CAL register contains the number of TFRO clock pulses in one tick. When the CALIB value is reached, one tick is issued. Setting 1 tick equal to 1 s is recommended. The default is 32,768 clock pulses in one tick.

Table 120. RTC timer calibration register (address 0x4005 4008) bit description

Bit	Symbol	Description	Reset value
15:0	CALIB	this value indicates the number of TFRO clock pulses in one tick	8000h
31:16	-	reserved	0

12.4.4 RTC timer sleep time register (SLEEPT)

The SLEEPT register is set to the time in ticks the timer has to sleep. The timer unit decrements this value each tick and the RTC interrupt is generated whenever the value reaches zero. Writing this register triggers a start unless the AUTOSTART bit is cleared. Manual start or restart is possible by writing logic 1 to CR[2].

Table 121. RTC timer sleep time register (address 0x4005 400C) bit description

Bit	Symbol	Description	Reset value
23:0	SLEEPTIME	this value indicates the time to sleep in ticks	0
31:24	-	reserved	0

12.4.5 RTC timer current value register (VAL)

The RTCVAL register contains the current remaining value of the timer in ticks.

Table 122. RTC timer current value register (address 0x4005 4010) bit description

Bit	Symbol	Description	Reset value
23:0	TIMEREMAIN	current value (remaining time) of the countdown timer	0
31:24	-	reserved	0

12.4.6 RTC timer interrupt mask set/clear register (IE)

This register controls whether the interrupt condition is enabled.

Note: *ARM uses the word ‘masked’ in the opposite sense from classic computer terminology, in which 'masked' meant ‘disabled’. ARM uses the word 'masked' to mean ‘enabled’. To avoid confusion, we do not use the word ‘masked’.*

Table 123. IMSC register (address 0x4005 4014) bit description

Bit	Symbol	Description	Reset value	Access
0	IE	software should set this bit to enable interrupt when the countdown is finished	0	R/W
31:1	-	reserved	0	-

12.4.7 RTC timer raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless of whether the interrupt is enabled in the IMSC register.

Table 124. RIS register (address 0x4005 4018) bit description

Bit	Symbol	Description	Reset value	Access
0	RIS	this bit is logic 1 when the countdown has finished	0	R
31:1	-	reserved	0	-

12.4.8 RTC timer masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the IMSC registers. When an RTC interrupt occurs, the interrupt service routine should read this register to determine the cause or causes of the interrupt.

Table 125. MIS register (address 0x4005 401C) bit description

Bit	Symbol	Description	Reset value	Access
0	MIS	this bit is logic 1 when the RTC countdown has finished and this interrupt is enabled	0	R
31:1	-	reserved	0	-

12.4.9 RTC timer interrupt clear register (ICR)

Software can write one or more logic ones to this write-only register, to clear the corresponding interrupt condition in the RTC timer.

Note: *The interrupt conditions can be disabled by clearing the corresponding bit in IMSC registers.*

Table 126. ICR register (address 0x4005 4020) bit description

Bit	Symbol	Description	Reset value	Access
0	IC	writing a logic 1 clears the countdown done interrupt	0	W
31:1	-	reserved	0	-

12.4.10 RTC timer access status register (ACCSTAT)

The access status register indicates whether the RTC is able to accept data over the APB. The RTC runs at a different clock rate than the core system, and bus synchronization is necessary. This mechanism is described below:

Write access

The write procedure is as follows:

- When the ARM writes to an RTC register:
  - The address/value pair is stored in the command-buffer
  - The direction flag is set to write
  - ACCCSTAT goes to logic 0 indicating a command is pending. This transition happens on an edge of the APB clock, which is faster than the TFRO clock
- When the TFRO clock edge comes, the address/value in the command-buffer is written to the RTC block
- On a next TFRO edge, the ACCSTAT goes to logic 0 to indicate that no command is pending and a new one can be given

Read access

The read procedure is as follows:

- When a value is stored in the PMU register for the ARM to read:
  - only the register address is stored in the command-buffer
  - the direction flag is set to read
  - ACCSTAT goes to logic 0 indicating a command is pending
- When the TFRO clock edge arrives, the address in the buffer is used to read the actual RTC register. This register value is copied in the (data part of the) command-buffer
- On the next TFRO edge, the ACCSTAT goes to logic 1 to indicate that the command is completed
- The command-buffer sees that the address of the second read is the same as the address part in the command-buffer. In this case, the result is not treated as a new read command ACCSTAT stays logic 1)
- However, the address part of the command-buffer is now cleared, so that a third read (even when from the same PMU register) causes a new command

Table 127. ACCSTAT register (address 0x4005 4024) bit description

Bit	Symbol	Description	Reset value	Access
0	READY	logic 0: RTC access not possible, logic 1: RTC access possible	0	R
31:1	-	reserved	0	-

12.4.11 RTC timer time register (TIME)

The RTC time register contains a counter which is incremented at each timer tick (RTC must be clocked and enabled). A value can be written to this register by the user.

Table 128. TIME register (address 0x4005 4030) bit description

Bit	Symbol	Description	Reset value	Access
31:0	TIME	current timer value	0	R/W

13 WatchDog Timer (WDT)

13.1 About this section

This section describes the watchdog timer, which is identical on all members of the NHS31xx family.

13.2 Purpose

If the microcontroller enters an erroneous state, the purpose of the watchdog-timer (WDT) is to reset the microcontroller within a predetermined length of time.

13.2.1 Configuration

The WDT is configured using the following:

- Pins:
  The WDT uses no external pins.
- Power:
  In the SYSAHBCLKCTRL register, set bit 11.
- Peripheral clock:
  Select the watchdog clock source and enable the WDT peripheral clock by writing to the WDTCLKDIV register.

13.2.2 Features

- Debug mode
- Enabled by software but requires a hardware reset or a watchdog reset/interrupt to be disabled
- If enabled, incorrect/incomplete feed sequence causes reset/interrupt
- Flag to indicate watchdog reset
- Programmable 24-bit timer with internal pre-scaler
- Selectable time period from (TWDCLK × 256 × 4) to (TWDCLK × 2^{24} × 4) in multiples of TWDCLK × 4
- The watchdog clock (WDCLK) source is the 8 MHz oscillator (SFRO)

13.2.3 General description

Register base address 0x4000 4000.

The watchdog consists of a divide-by-4 fixed prescaler and a 24-bit counter. The clock is fed to the timer via a prescaler. When clocked, the timer decrements. The minimum value from which the counter decrements is 0xFF. Setting a value lower than 0xFF causes 0xFF to be loaded in the counter. So, the minimum watchdog interval is (TWDCLK × 256 × 4) and the maximum is (TWDCLK × 2^{24} × 4), in multiples of (TWDCLK × 4).

Use the watchdog in the following way:

1. Set the watchdog timer constant reload value in WDTC register.
2. Set up the watchdog timer operating mode in WDMOD register.
3. Enable the watchdog by writing 0xAA followed by 0x55 to the WDFEED register.

4. To prevent a reset-interrupt, the watchdog must be fed again before the watchdog counter underflows.

When the WDT is in reset mode and the counter reaches zero, the CPU is reset and loads the stack pointer and program counter from the vector table. The same happens when an external reset occurs. To determine if the WDT has caused the reset condition, the WDT timeout flag (WDTOF) can be examined. Software must clear the WDTOF flag.

When the watchdog is enabled (but not in reset mode), the counter underflows, and the Arm M0+ core is in sleep mode or deep-sleep mode, the Arm wakes up.

The watchdog timer block uses two clocks: PCLK and WDCLK. PCLK is used for the APB accesses to the watchdog registers and is derived from the system clock (see Figure 7). The WDCLK is used for the watchdog timer counting and is derived from the wdt_clk. The SFRO is used as a clock source for wdt_clk clock. The WDCLK has its own clock divider which can also disable this clock.

There is some synchronization logic between these two clock domains. When APB operations update the WDMOD and WDTC registers, the new value takes effect in 3 WDCLK cycles on the logic in the WDCLK clock domain. When the watchdog timer is counting on WDCLK, the following procedure occurs: The synchronization logic locks the counter value on WDCLK and then synchronizes it with the PCLK for reading as the WDTV register by the CPU.

13.3 Register descriptions

The Watchdog timer uses the registers shown in Table 129. More detailed descriptions follow.

Table 129. Register overview: Watchdog timer block (base address 0x4000 4000)

Name	Access	Address offset	Description	Reset value	Reference
WDMOD	R/W	0x000	Watchdog mode register. This register contains the Basic mode and status of the Watchdog Timer	0x0000 0000	-
WDTC	R/W	0x004	watchdog timer constant register. This register determines the timeout value	0x0000 00FF	-
WDFEED	W	0x008	watchdog feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the Watchdog timer with the value contained in WDTC	-	-
WDTV	R	0x00C	watchdog timer value register. This register reads out the current value of the Watchdog timer	0x0000 00FF	-

13.3.1 Watchdog mode register

The WDMOD register controls the operation of the watchdog through the combination of WDEN and RESET bits.

Note: *A watchdog feed must be performed before any changes to the WDMOD register take effect.*

Table 130. WDMOD register (address 0x4000 4000) bit description

Bit	Symbol	Description	Reset value
0	WDEN	watchdog enable bit (set only). When logic 1, the watchdog timer is running	0
1	WDRESET	watchdog reset enable bit (set only). When logic 1, a watchdog timeout causes a chip reset	0
2	WDTOF	watchdog timeout flag. Set when the watchdog timer times out, cleared by software. (read/write)	0
3	WDINT	watchdog interrupt flag (read-only, not clearable by software)	0
7:4	-	reserved	-
31:8	-	reserved	-

When the WDEN and/or WDRESET bits are set, software cannot clear them. A reset or a watchdog timer underflow clears both flags.

WDTOF

The watchdog timeout flag is set when the watchdog times out. Software or a POR reset clears this flag.

WDINT

When the Watchdog times out, the watchdog interrupt flag is set. When any reset occurs, this flag is cleared. When the watchdog interrupt is serviced, it must be disabled in the NVIC or the watchdog interrupt request is generated indefinitely. The intent of the watchdog interrupt is to allow debugging watchdog activity without resetting the device when the watchdog overflows.

Watchdog reset or interrupt occurs anytime the watchdog is running and has an operating clock source. Any clock source works in sleep mode. If a watchdog interrupt occurs in sleep mode, it wakes the device.

Table 131. Watchdog operating mode selection

WDEN	WDRESET	Mode of operation
0		debug/operate without the watchdog running
1	0	Watchdog interrupt mode: debug with the watchdog interrupt but no WDRESET enabled. When this mode is selected, a watchdog counter-underflow sets the WDINT flag and the watchdog interrupt request is generated. ***Note:*** *In interrupt mode, check the WDINT flag. If this flag is set, the interrupt is true and the interrupt routine can service it. If this flag is not set, the interrupt should be ignored.*

Table 131. Watchdog operating mode selection...*continued*

WDEN	WDRESET	Mode of operation
1	1	Watchdog reset mode: operate with the watchdog interrupt and WDRESET enabled. When this mode is selected, a watchdog counter underflow resets the microcontroller. Although the watchdog interrupt is also enabled in this case (WDEN = 1), it is not recognized because the watchdog reset clears the WDINT flag.

13.3.2 Watchdog timer constant register

The WDTC register determines the timeout value. Every time a feed sequence occurs the WDTC content is reloaded in to the watchdog timer. It is a 32-bit register with 8 LSB set to logic 1 on reset. Writing values below 0xFF cause 0x0000 00FF to be loaded to the WDTC. So, the minimum timeout interval is $T_{WDCLK} \times 256 \times 4$.

Table 132. WDTC register (address 0x4000 4004) bit description

Bit	Symbol	Description	Reset value
23:0	Count	Watchdog timeout interval	0x0000 00FF
31:24	-	reserved	0

13.3.3 Watchdog feed register

Writing 0xAA followed by 0x55 to this register reloads the watchdog timer with the WDTC value. If this operation is enabled via the WDMOD register, it also starts the watchdog. Setting the WDEN bit in the WDMOD register is not sufficient to enable the watchdog. A valid feed sequence must be completed after setting WDEN before the watchdog can generate a reset. Until then, the watchdog ignores feed errors. After writing 0xAA to WDFEED, access to any watchdog register other than writing 0x55 to WDFEED causes an immediate reset/interrupt when the watchdog is enabled. The reset is generated during the second PCLK following incorrect access to a watchdog register during a feed sequence.

Interrupts must be disabled during the feed sequence. If an interrupt happens during the feed sequence, an abort condition occurs.

Table 133. WDFEED register (address 0x4000 4008) bit description

Bit	Symbol	Description	Reset value
0:7	Feed	feed value must be 0xAA followed by 0x55	-
8:31	-	reserved	-

13.3.4 Watchdog timer value register

The WDTV register is used to read the current value of watchdog timer. When reading the value of the 24-bit timer, the lock and synchronization procedure takes up to 6 WDCLK cycles plus 6 PCLK cycles. The result of this procedure is that the value of WDTV is older than the actual value of the timer when the CPU is reads it.

Table 134. WDTV register (address 0x4000 400C) bit description

Bit	Symbol	Description	Reset value
23:0	Count	counter-timer value	0x0000 00FF
31:24	-	reserved	-

14 SYSTICK timer

14.1 About this section

This section describes the SYSTICK timer, which is identical on all members of the NHS31xx family.

14.2 Purpose

The SYSTICK timer is a part of the Arm Cortex-M0+. The SYSTICK timer is intended to generate a fixed 10 millisecond interrupt that an operating system or other system management software can use.

The SYSTICK timer is configured using the following registers:

- **Pins:**
  The system tick timer uses no external pins.
- **Power:**
  The system tick timer is enabled through the SYSTICK control register. The system tick timer clock is fixed to half the frequency of the system clock.
- Enable the clock source for the SYSTICK timer in the CSR register

14.3 Features

- Simple 24-bit timer
- Uses dedicated exception vector
- Clocked internally by the system clock or the system clock ÷ 2

14.4 General description

The block diagram of the SYSTICK timer is shown Figure 22.

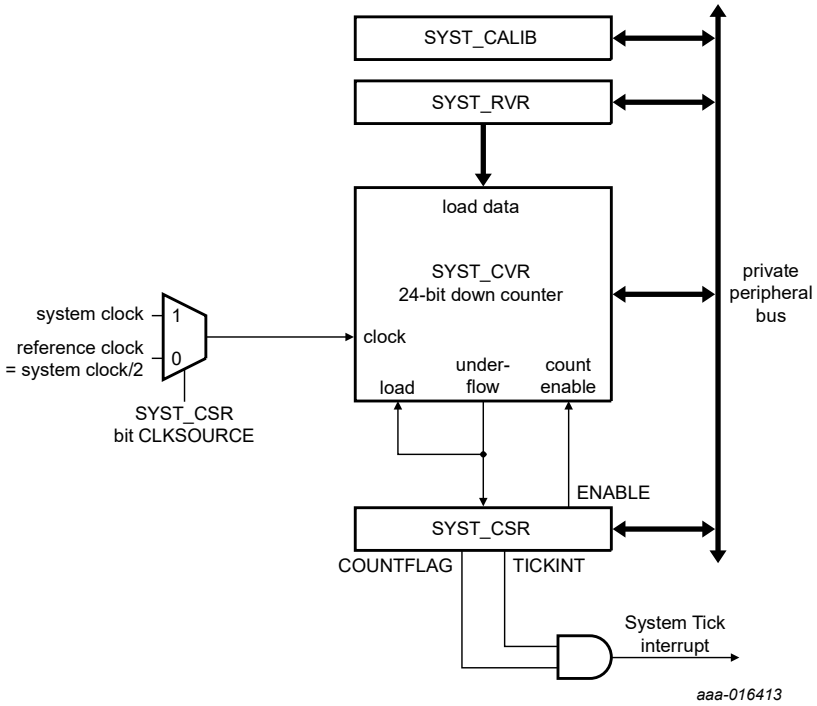


Figure 22. System tick timer block diagram

The SYSTICK timer is a part of the Cortex-M0+. It is intended to generate a fixed 10 millisecond interrupt for an operating system or other system management software can use. It also facilitates porting of software by providing a standard timer that is available on Cortex-M0-based devices.

The SYSTICK timer can be used for:

- An RTOS tick timer which fires at a programmable rate (for example, 100 Hz) and invokes a SYSTICK routine
- A high-speed alarm timer using the core clock
- A simple counter
  Software can use this counter to measure time to completion and time used.
- It can be determined if an action completed within a set duration by reading the COUNTFLAG bit field in the control and status register

14.5 Functional description

The SYSTICK timer is a 24-bit timer that counts down to zero and generates an interrupt. The intent is to provide a fixed 10 millisecond time interval between interrupts. The SYSTICK timer is clocked from the CPU system clock (see Figure 22) or from the reference clock. The reference clock is fixed to half the frequency of the CPU clock. To generate recurring interrupts at a specific interval, the RVR register must be initialized with the correct value for the desired interval. A default value is provided in the CALIB register which is software can change.

If the CPU clock is set to 2 MHz, this frequency gives a 10 millisecond interrupt rate.

14.5.1 Example timer calculation

To use the system tick timer, do the following:

1. Program the RVR register with the reload value RELOAD to obtain the desired time interval.
2. Clear the CVR register by writing to it. This action ensures that the timer counts from the RVR value rather than an arbitrary value when the timer is enabled.
3. Program the SCR register with the value 0x7 which enables the SYSTICK timer and the SYSTICK timer interrupt.

The following example illustrates the selection of the timer reload value to obtain a 10 ms time interval with the system clock set to 20 MHz.

The system tick clock = system clock = 8 MHz. Bit CLKSOURCE in the CSR register set to 1 (system clock).

RELOAD = (system tick clock
frequency × 10 ms) − 1 = (8 MHz × 10 ms) − 1 = 80000 − 1 = 79999 = 0x0001 387F.

14.6 Register descriptions

The SYSTICK timer registers are on the Arm Cortex-M0+ private peripheral bus. They are part of the Arm Cortex-M0+ core peripherals.

Table 135. Register overview: SYSTICK timer (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value [1]
CSR	R/W	0x010	system timer control and status Register	0x0000 0000
RVR	R/W	0x014	system timer reload value register	0x0000 4E1F
CVR	R/W	0x018	system timer current value register	0x0000 0000
CALIB	R/W	0x01C	system timer calibration value register	0x0000 0004

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

14.6.1 System timer control and status register (CSR)

The CSR register contains control information for the SYSTICK timer and provides a status flag. This register is part of the Arm Cortex-M0+ core system timer register block.

This register determines the clock source for the system tick timer.

Table 136. CSR register (address 0xE000 E010) bit description

Bit	Symbol	Description	Reset value
0	ENABLE	enable system tick counter. When logic 1, the counter is enabled. When logic 0, the counter is disabled.	0
1	TICKINT	system tick interrupt enable. When logic 1, the system tick interrupt is enabled. When logic 0, the system tick interrupt is disabled. When enabled, the interrupt is generated when the system tick counter counts down to 0	0
2	CLKSOURCE	system tick clock source selection. When logic 1, the system clock (CPU) clock is selected. When logic 0, the system clock/2 is selected as the reference clock.	0
15:3	-	reserved - do not write logic ones. The value read from a reserved bit is not defined.	-
16	COUNTFLAG	returns 1 if the SYSTICK timer counted to logic 0 since the last read of this register.	0
31:17	-	reserved. Do not write logic ones. The value read from a reserved bit is not defined.	-

14.6.2 System timer reload value register (RVR)

The RVR register is set to the value that is loaded into the SYSTICK timer whenever it counts down to zero. Timer initialization loads this register via software. If the CPU is running at the frequency intended for use with the CALIB value, the CALIB register may be read and used as the value for RVR register.

Table 137. RVR register (address 0xE000 E014) bit description

Bit	Symbol	Description	Reset value
23:0	RELOAD	This value that is loaded into the system tick counter when it counts down to 0.	0xFFFFFF
31:24	-	reserved. Do not write logic ones. The value read from a reserved bit is not defined.	-

14.6.3 System timer current value register (CVR)

The CVR register returns the current count from the system tick counter.

Table 138. CVR register (address 0xE000 E018) bit description

Bit	Symbol	Description	Reset value
23:0	CURRENT	reading this register returns the current value of the system tick counter. Writing any value clears the system tick counter and the COUNTFLAG bit in STCTRL.	0xFFFFFF
31:24	-	reserved. Do not write logic ones. The value read from a reserved bit is not defined.	-

14.6.4 System timer calibration value register (CALIB)

The value of the SYSTCKCAL register in the system configuration block SYSCON (see Table 23) drives the value of the CALIB register.

Table 139. CALIB register (address 0xE000 E01C) bit description

Bit	Symbol	Description	Reset value
23:0	TENMS	10 ms calibration value	0x4
29:24	-	reserved. Do not write logic ones. The value read from a reserved bit is not defined.	-
30	SKEW	if logical 1, the TENMS bit field is not accurate.	0
31	NOREF	if logical 1, it indicates that SYSTICK uses the core clock for counting as no external reference clock is available.	1’b0

15 16-bit timer CT16B

15.1 About this section

This section describes the CT16B 16-bit timer, which is identical on all members of the NHS31xx family.

15.2 Configuration

The CT16B is configured as follows:

- **Pins:**
  The match outputs of the timer can be routed to I/O pins by setting the I/O pin function in the IOCON register (see Table 254).
- **Clock:**
  The PCLK clock clocks the CT16B block. This clock is derived from the system clock (see Figure 7). The clock can be enabled by setting bit 8 in the SYSAHBCLKCTRL register (see Table 17).

15.3 Features

One 16-bit timer with a programmable 16-bit prescaler.

- Timer operation
- Four 16-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match
  - Stop timer on match with optional interrupt generation
  - Reset timer on match with optional interrupt generation
- Up to two CT16B external outputs corresponding to the match registers with the following capabilities:
  - Set LOW on match
  - Set HIGH on match
  - Toggle on match
  - Do nothing on match
- Up to two match registers can be configured as PWM allowing the use of up to two match outputs as single edge controlled PWM outputs.

15.4 Applications

- Interval timer for generating internal events
- Free-running timer
- Pulse width modulator via match outputs

15.5 General description

The timer is designed to count cycles of the peripheral clock (PCLK). Optionally, it can also generate interrupts or perform other actions at specified timer values based on four match registers. The system clock (see Figure 7) provides the peripheral clock.

In PWM mode, four match registers can be used to provide a single-edge controlled PWM output on the match output pins. Use of the match registers that are not pinned out to control the PWM cycle length is recommended.

15.6 Pin description

Table 140 gives a brief summary of each of the timer-related pins.

Table 140. Counter/timer pin description

Pin	Type	Description
CT16B_M0	output	16-bit timer-match output 0
CT16B_M1	output	16-bit timer-match output 1

15.7 Example timer operation

Figure 23 shows a timer configured to reset the count and generate an interrupt on match. The prescaler is set to 2 and the match register set to 6. At the end of the timer cycle, where the match occurs, the timer count is reset giving a full length cycle to the match value. The match interrupt is generated in the next clock after the timer reached the match value.

Figure 24 shows a timer configured to stop and generate an interrupt on match. The prescaler is again set to 2 and the match register set to 6. The match interrupt is generated in the next clock after the timer reached the match value. The timer enable bit in TCR is cleared.

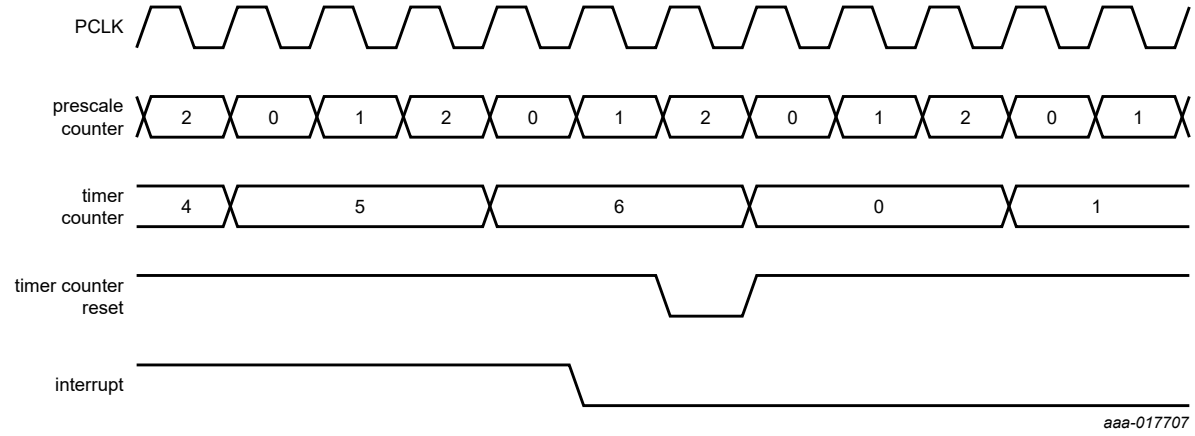


Figure 23. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled

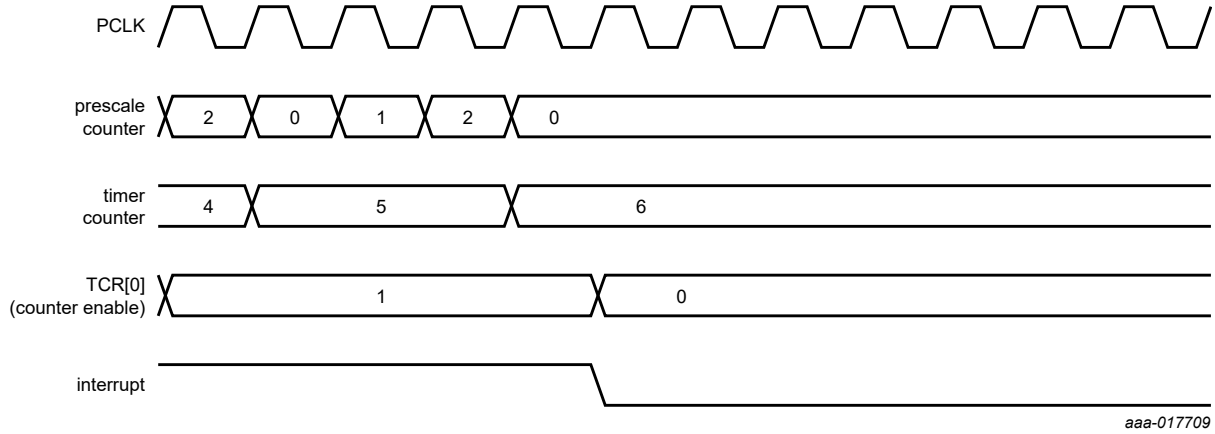


Figure 24. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

15.8 Architecture

Figure 25 shows the block diagram for timer0 and timer1.

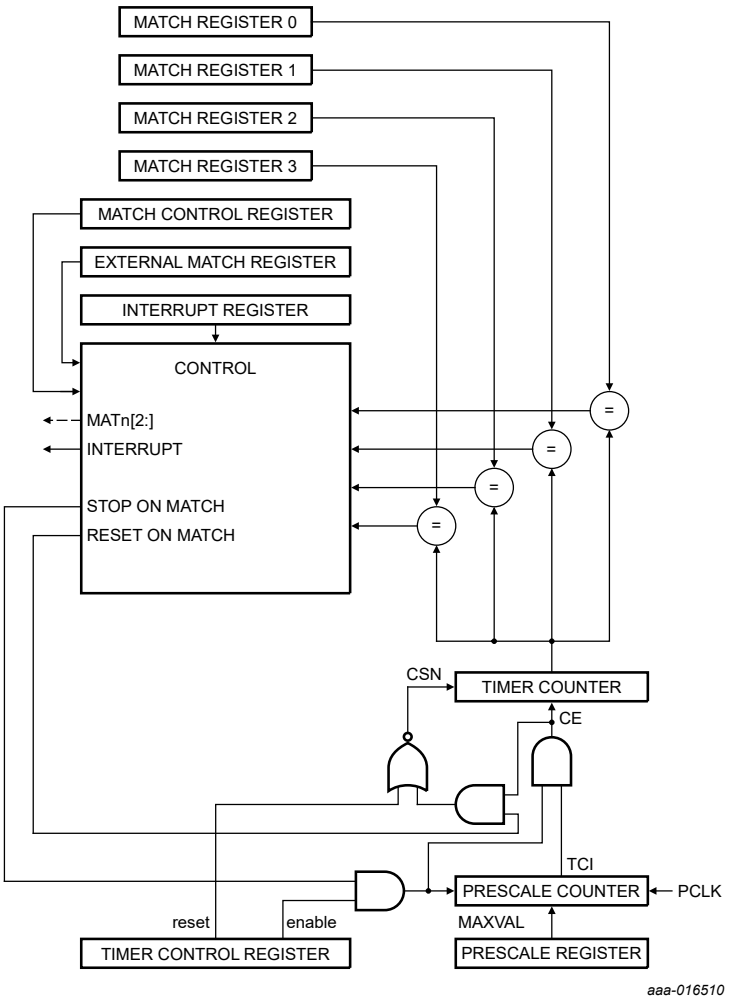


Figure 25. 16-bit timer block diagram

15.9 Register descriptions

The 16-bit timer contains the registers shown in Table 141. More detailed descriptions follow.

Table 141. Register overview: 16-bit timer 0 CT16B0 (base address 0x4000 C000)

Name	Access	Address offset	Description	Reset value [1]
IR	R/W	0x000	Interrupt Register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending	0x0000 0000
TCR	R/W	0x004	Timer Control Register (TCR). The TCR is used to control the timer functions. The Timer can be disabled or reset through the TCR	0x0000 0000
TC	R/W	0x008	Timer (TC). The 16-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR	0x0000 0000
PR	R/W	0x00C	Prescale Register (PR). When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC	0x0000 0000
PC	R/W	0x010	Prescale Counter (PC). The 16-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface	0x0000 0000
MCR	R/W	0x014	Match Control Register (MCR). The MCR is used to control whether an interrupt is generated and if the TC is reset when a Match occurs	0x0000 0000
MR0	R/W	0x018	Match Register 0 (MR0). Enabled MR0 through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC	0x0000 0000
MR1	R/W	0x01C	Match Register 1 (MR1). See MR0 description	0x0000 0000
MR2	R/W	0x020	Match Register 2 (MR2). See MR0 description	0x0000 0000
MR3	R/W	0x024	Match Register 3 (MR3). See MR0 description	0x0000 0000
-	-	0x028 - 0x038	reserved	-
EMR	R/W	0x03C	External Match Register (EMR). The EMR controls the match function and the external match pins CT16B0_M[1:0]	0x0000 0000
-	-	0x040 - 0x070	reserved	-
PWMC	R/W	0x074	PWM Control Register (PWMC). The PWMC enables PWM mode for the external match pins CT16B0_M[1:0]	0x0000 0000

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

15.9.1 Interrupt register (IR)

The Interrupt register (IR) consists of 4 bits for the match interrupts and 1 bit for the capture interrupt. If an interrupt is generated, the corresponding bit in the IR is HIGH. Otherwise, the bit is LOW. Writing a logic 1 to the corresponding IR bit resets the interrupt. Writing a logic 0 has no effect.

Table 142. IR register (address 0x4000 C000) bit description

Bit	Symbol	Description	Reset value
0	MR0INT	interrupt flag for match channel 0	0
1	MR1INT	interrupt flag for match channel 1	0
2	MR2INT	interrupt flag for match channel 2	0
3	MR3INT	interrupt flag for match channel 3	0
31:4	-	reserved	-

15.9.2 Timer control register (TCR)

The timer control register (TCR) is used to control the operation of the timer.

Table 143. TCR register (address 0x4000 C004) bit description

Bit	Symbol	Description	Reset value
0	CEN	Timer enable. When logic 1, the timer counter and prescale counter are enabled for counting. When logic 0, the counters are disabled.	0
1	CRST	Timer reset. When logic 1, the timer counter and the prescale counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to logic 0.	0
2:31	-	reserved	-

15.9.3 Timer counter register (TC)

When the prescale counter reaches its terminal count, the 16-bit timer counter is incremented. Unless reset before reaching its upper limit, the TC counts up to 0x0000 FFFF and then wraps back to the value 0x0000 0000. This event does not cause an interrupt, but a match register can be used to detect possible overflow, if necessary.

Table 144. TC register (address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
15:0	TC	Timer-counter value	0
31:16	-	reserved	-

15.9.4 Prescale register (PR)

The 16-bit prescale register specifies the maximum value for the prescale counter.

Table 145. PR register (address 0x4000 C00C) bit description

Bit	Symbol	Description	Reset value
15:0	PR	Prescale maximum value	0
31:16	-	reserved	-

15.9.5 Prescale counter register (PC)

The 16-bit prescale counter controls division of PCLK by some constant value before it is applied to the timer counter. This division allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows. The prescale counter is incremented on every PCLK. When it reaches the value stored in the prescale register, the timer counter is incremented and the prescale counter is reset on the next PCLK. This reset causes the TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, and so on.

Table 146. PC register (address 0x4001 C010) bit description

Bit	Symbol	Description	Reset value
15:0	PC	Prescale counter value	0
31:16	-	reserved	-

15.9.6 Match control register (MCR)

The match control register is used to control what operations are performed when one of the match registers matches the timer counter. Table 147 shows the function of each of the bits.

Table 147. MCR register (address 0x4000 C014) bit description

Bit	Symbol	Description		Reset value
0	MR0I	interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC		0
		0	disabled	
		1	enabled	
1	MR0R	reset on MR0: If MR0 matches the TC, then the TC is reset		0
		0	disabled	
		1	enabled	
2	MR0S	stop on MR0: If MR0 matches the TC, then the TC and PC are stopped and TCR[0] is set to logic 0		0
		0	disabled	
		1	enabled	
3	MR1I	interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC		0
		0	disabled	
		1	enabled	

Table 147. MCR register (address 0x4000 C014) bit description*...continued*

Bit	Symbol	Description		Reset value
4	MR1R	reset on MR1: If MR1 matches the TC, then the TC is reset		0
		0	disabled	
		1	enabled	
5	MR1S	stop on MR1: If MR1 matches the TC, then the TC and PC are stopped and TCR[0] is set to logic 0		0
		0	disabled	
		1	enabled	
6	MR2I	interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC		0
		0	disabled	
		1	enabled	
7	MR2R	reset on MR2: If MR2 matches the TC, then, the TC is reset		0
		0	disabled	
		1	enabled	
8	MR2S	stop on MR2: If MR2 matches the TC, then the TC and PC are stopped and TCR[0] is set to logic 0		0
		0	disabled	
		1	enabled	
9	MR3I	interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC		0
		0	disabled	
		1	enabled	
10	MR3R	reset on MR3: If MR3 matches the TC, then the TC is reset		0
		0	disabled	
		1	enabled	
11	MR3S	stop on MR3: If MR3 matches the TC, then the TC and PC are stopped and TCR[0] is set to logic 0		0
		0	disabled	
		1	enabled	
31:12	-	reserved		-

15.9.7 Match register (MR0/1/2/3)

The match register values are continuously compared to the counter/timer value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the counter/timer, or stop the timer. The settings in the MCR register control the actions.

Table 148. MR0/1/2/3 registers (addresses 0x4000 C018 to 24) bit description

Bit	Symbol	Description	Reset value
15:0	MATCH	Timer-Counter match value	0

Table 148. MR0/1/2/3 registers (addresses 0x4000 C018 to 24) bit description...*continued*

Bit	Symbol	Description	Reset value
31:16	-	reserved	-

15.9.8 External match register (EMR)

The external match register provides both control and status of the external match channels and external match pins CT16B_MAT[1:0].

If the match outputs are configured as PWM outputs in the PWMCON registers, PWM rules (see Section 15.9.10) determine the function of the external match registers.

Table 149. EEMR register (address 0x4000 C03C) bit description

Bit	Symbol	Description		Reset value
0	EM0	External match 0. This bit reflects the state of output CT16B_M0, regardless of this output being connected to its pin or not. When PWM channel 0 is enabled, a match between TC and MR0 raises the output pin high. Bits EMR[5:4] control the functionality of this output. If the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH), this bit is driven to the CT16B_M0 pins.		0
1	EM1	External match 1. This bit reflects the state of output CT16B_M1, regardless of this output being connected to its pin or not. When PWM channel 1 is enabled, a match between TC and MR1 raises the output pin high. Bits EMR[7:6] control the functionality of this output. This bit is driven to the CT16B_M1 pins.		0
3:2	-	reserved		-
5:4	EMC0	External match control 0. Determines the functionality of external match 0.		0
		00b	do nothing.	
		01b	clear the corresponding external match bit/output to logic 0 (CT16B_M0 pin is LOW if pinned out).	
		10b	set the corresponding external match bit/output to logic 1 (CT16B_M0 pin is HIGH if pinned out).	
		11b	toggle the corresponding external match bit/output. ***Note:*** *The toggle only works when the corresponding match register is higher than 0.*	
7:6	EMC1	external match control 1. Determines the functionality of external match 1.		0
		00b	do nothing.	
		01b	clear the corresponding external match bit/output to logic 0 (CT16B_M1 pin is LOW if pinned out).	
		10b	set the corresponding external match bit/output to logic 1 (CT16B_M1 pin is HIGH if pinned out).	
		11b	toggle the corresponding external match bit/output. ***Note:*** *The toggle only works when the corresponding match register is higher than 0.*	

15.9.9 PWM control register (PWMC)

The PWM control register is used to configure the match outputs as PWM outputs. Each match output can be independently set as either a PWM output, or match output. The external match register (EMR) controls the function of the match output.

Three single-edge controlled PWM outputs can be selected on the CT16B_M[1:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The match register that is configured to set the PWM cycle length, resets the timer. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Table 150. PWMC register (address 0x4000 C074) bit description

Bit	Symbol	Description		Reset value
0	PWMEN0	PWM channel0 enable		0
		0	EM0 controls the CT16B_M0	
		1	PWM mode is enabled for CT16B_M0	
1	PWMEN1	PWM channel1 enable		0
		0	EM1 controls the CT16B0_M1	
		1	PWM mode is enabled for CT16B_M1	
31:2	-	reserved		-

15.9.10 Rules for single-edge controlled PWM outputs

1. All single-edge controlled PWM outputs go LOW at the beginning of each PWM cycle unless their match value is equal to zero.
2. When its match value is reached, each PWM output goes HIGH. If no match occurs (that is, the match value is greater than the PWM cycle length), the PWM output remains continuously LOW.
3. If a match value higher than the PWM cycle length is written to the match register and the PWM signal is HIGH already, the PWM signal is cleared at the start of the next PWM cycle.
4. If a match register value equals the timer reset value, the PWM output is reset to LOW on the next clock tick. So, the PWM output always consists of a one clock tick wide positive pulse. The PWM cycle length is the same as the timer reset value.
5. If a match register is set to zero, the PWM output goes HIGH the first time the timer goes back to zero and remains HIGH

Note: *When the match outputs are used as PWM outputs, the reset and stop bits in the match control register (MCR) must be set to logic 0, except for the match register. The match register sets the PWM cycle length. For this register, set the MRR bit to logic 1 to enable the timer reset when the timer value matches the value of the corresponding match register.*

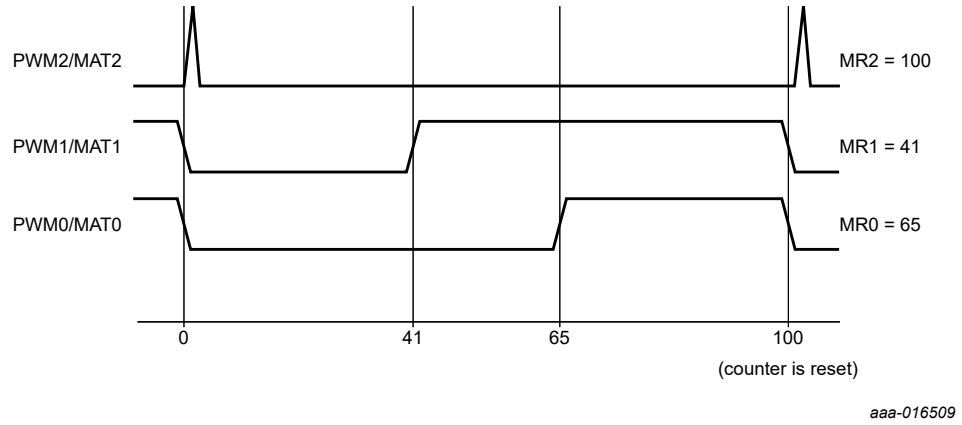


Figure 26. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR2) and MAT2:0 enabled as PWM outputs by the PWCM register

16 32-bit timer CT32B

16.1 About this section

This section describes the CT32B 32-bit timer, which is identical on all members of the NHS31xx family.

16.2 Configuration

The CT32B is configured as follows:

- **Pins:**
  The match outputs of the timer can be routed to I/O pins by setting the I/O pin function in the IOCON register (see Table 254).
- **Clock:**
  The PCLK clock clocks the CT32B block. This clock is derived from the system clock (see Figure 7). The clock can be enabled by setting bit 8 in the SYSAHBCLKCTRL register (see Table 17).

16.3 Features

One 32-bit timer with a programmable 32-bit prescaler.

- Timer operation
- Four 32-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match
  - Stop timer on match with optional interrupt generation
  - Reset timer on match with optional interrupt generation
- Up to two CT32B external outputs corresponding to the match registers with the following capabilities:
  - Set LOW on match
  - Set HIGH on match
  - Toggle on match
  - Do nothing on match
- Up to two match registers can be configured as PWM allowing the use of up to two match outputs as single edge controlled PWM outputs.

16.4 Applications

- Interval timer for generating internal events
- Free-running timer
- Pulse width modulator via match outputs

16.5 General description

The timer is designed to count cycles of the peripheral clock (PCLK). Optionally, It can also generate interrupts or perform other actions at specified timer values based on four match registers. The system clock (see Figure 7) provides the peripheral clock.

In PWM mode, four match registers can be used to provide a single-edge controlled PWM output on the match output pins. Using the match registers that are not pinned out to control the PWM cycle length is recommended.

16.6 Pin description

Table 151 gives a brief summary of each of the counter/timer related pins.

Table 151. Counter/timer pin description

Pin	Type	Description
CT32B_M0	output	32-bit timer-match output 0
CT32B_M1	output	32-bit timer-match output 1

16.7 Example timer operation

Figure 27 shows a timer configured to reset the count and generate an interrupt on match. The prescaler is set to 2 and the match register set to 6. At the end of the timer cycle, where the match occurs, the timer count is reset giving a full length cycle to the match value. The match interrupt is generated in the next clock after the timer reached the match value.

Figure 28 shows a timer configured to stop and generate an interrupt on match. The prescaler is again set to 2 and the match register set to 6. The match interrupt is generated in the next clock after the timer reached the match value. The timer enable bit in TCR is cleared.

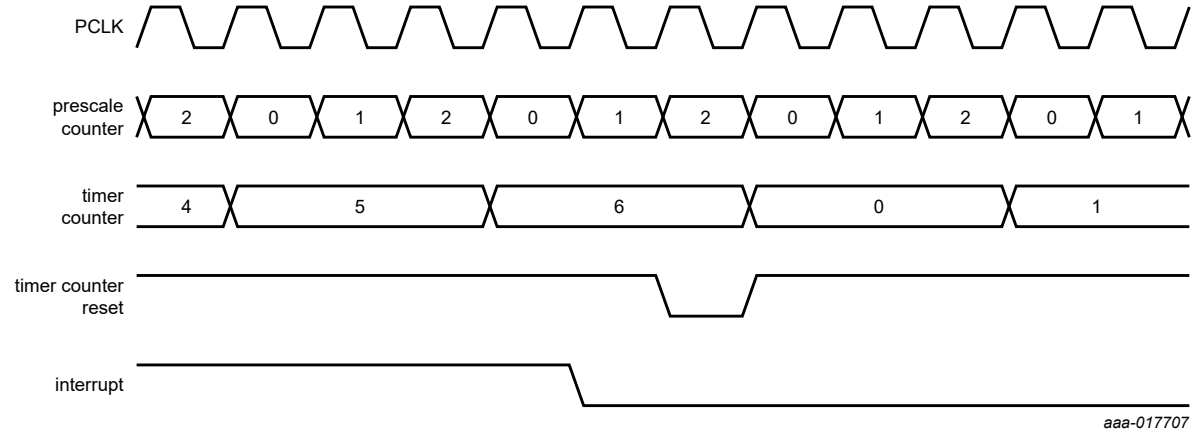


Figure 27. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled

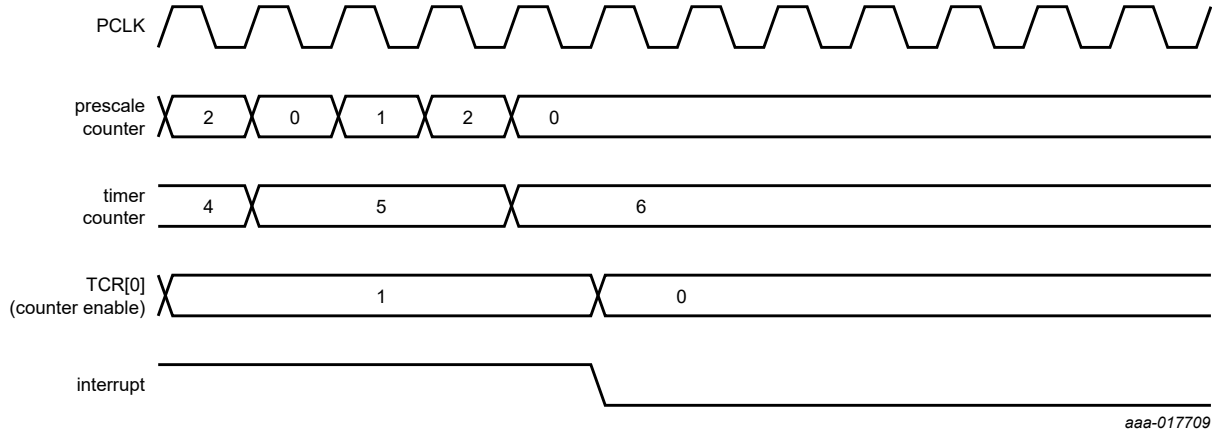


Figure 28. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

16.8 Architecture

Figure 29 shows the block diagram for counter/timer0 and counter/timer1.

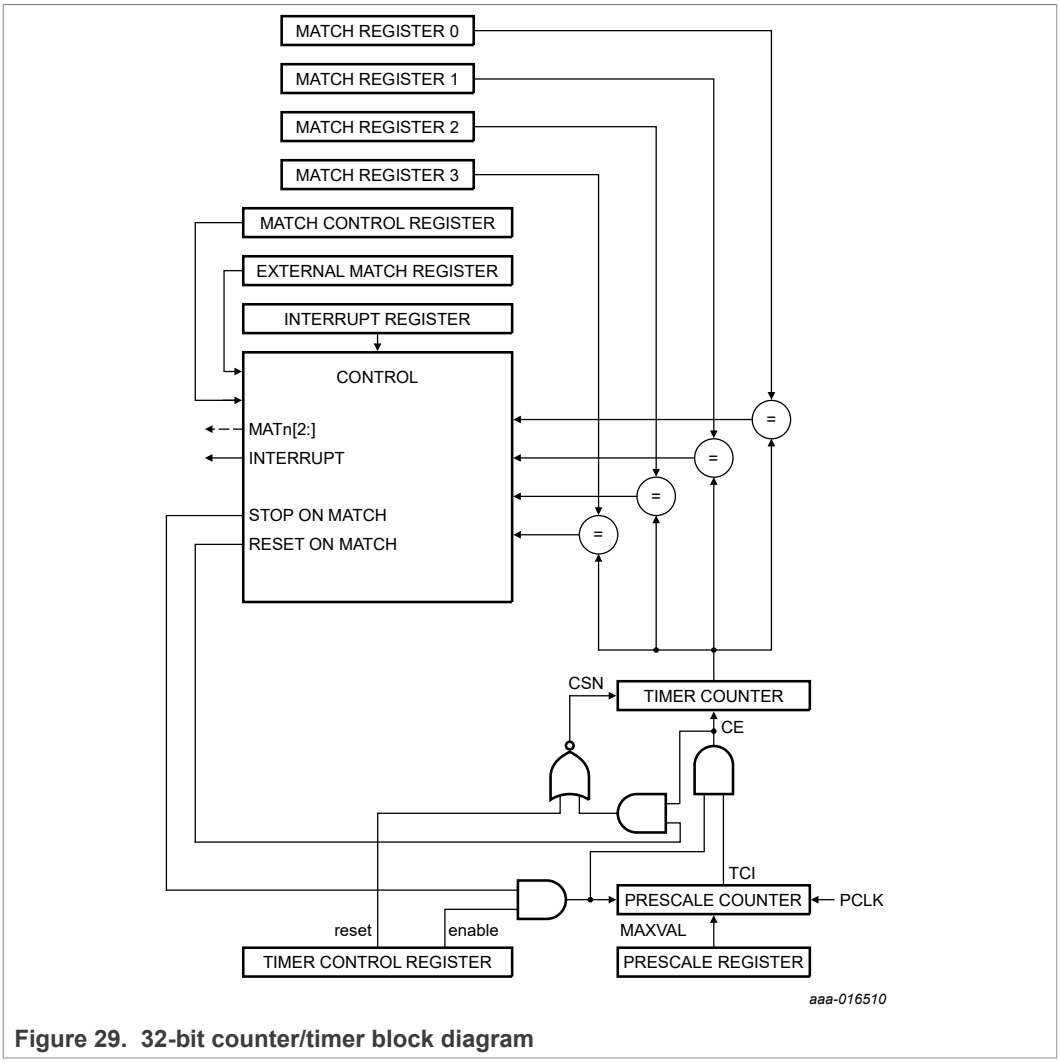


Figure 29. 32-bit counter/timer block diagram

16.9 Register descriptions

The 32-bit counter/timer contains the registers shown in Table 152. More detailed descriptions follow.

Table 152. Register overview: 32-bit counter/timer 0 CT32B (base address 0x4000 4000)

Name	Access	Address offset	Description	Reset value [1]
IR	R/W	0x000	Interrupt register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending.	0x0000 0000
TCR	R/W	0x004	Timer control register (TCR). The TCR is used to control the timer counter functions. The timer counter can be disabled or reset through the TCR.	0x0000 0000
TC	R/W	0x008	Timer counter (TC). The 32-bit TC is incremented every PR + 1 cycles of PCLK. The TC is controlled through the TCR.	0x0000 0000
PR	R/W	0x00C	Prescale register (PR). When the prescale counter (below) is equal to this value, the next clock increments the TC and clears the PC.	0x0000 0000
PC	R/W	0x010	Prescale counter (PC). The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incriminated and the PC is cleared. The PC is observable and controllable through the bus interface.	0x0000 0000
MCR	R/W	0x014	Match control register (MCR). The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	0x0000 0000
MR0	R/W	0x018	Match register 0 (MR0). Enabled MR0 through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC	0x0000 0000
MR1	R/W	0x01C	Match register 1 (MR1). See MR0 description	0x0000 0000
MR2	R/W	0x020	Match register 2 (MR2). See MR0 description	0x0000 0000
MR3	R/W	0x024	Match register 3 (MR3). See MR0 description	0x0000 0000
-	R/W	0x028 - 0x038	reserved	-
EMR	R/W	0x03C	External match register (EMR). The EMR controls the match function and the external match pins CT326B_M[1:0].	0x0000 0000
-	-	0x040 - 0x070	reserved	-
PWMC	R/W	0x074	PWM control register (PWMC). The PWMC enables PWM mode for the external match pins CT32B_M[1:0].	0x0000 0000

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

16.9.1 Interrupt register (IR)

The interrupt register (IR) consists of 4 bits for the match interrupts and 1 bit for the capture interrupt. If an interrupt is generated, the corresponding bit in the IR is HIGH. Otherwise, the bit is LOW. Writing a logic 1 to the corresponding IR bit resets the interrupt. Writing a logic 0 has no effect.

Table 153. IR register (address 0x4000 4000) bit description

Bit	Symbol	Description	Reset value
0	MR0INT	interrupt flag for match channel 0	0x0000 0000
1	MR1INT	interrupt flag for match channel 1	0x0000 0000
2	MR2INT	interrupt flag for match channel 2	0x0000 0000
3	MR3INT	interrupt flag for match channel 3	0x0000 0000
31:4	-	reserved	-

16.9.2 Timer control register (TCR)

The timer control register (TCR) is used to control the operation of the counter/timer.

Table 154. TCR register (address 0x4000 4004) bit description

Bit	Symbol	Description	Reset value
0	CEN	Counter enable. When one, the timer counter and the prescale counter are enabled for counting. When zero, the counters are disabled.	0
1	CRST	Counter reset. When one, the timer counter and the prescale counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
31:2	-	reserved	-

16.9.3 Timer counter register (TC)

When the prescale counter reaches its terminal count, the 32-bit timer counter is incremented. Unless reset before reaching its upper limit, the TC counts up to 0xFFFF FFFF and then wraps back to the value 0x0000 0000. This event does not cause an interrupt, but a match register can be used to detect possible overflow if necessary.

Table 155. TC register (address 0x4000 4008) bit description

Bit	Symbol	Description	Reset value
31:0	TC	Timer Counter value	0

16.9.4 Prescale register (PR)

The 32-bit prescale register specifies the maximum value for the prescale counter.

Table 156. PR register (address 0x4000 400C) bit description

Bit	Symbol	Description	Reset value
31:0	PR	Prescale maximum value	0

16.9.5 Prescale counter register (PC)

The 32-bit prescale counter controls division of PCLK by some constant value before it is applied to the timer counter. This division allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows. The prescale counter is incremented on every PCLK. When it reaches the value stored in the prescale register, the timer counter is incremented and the prescale counter is reset on the next PCLK. This reset causes the TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, and so on.

Table 157. PC register (address 0x4001 4010) bit description

Bit	Symbol	Description	Reset value
31:0	PC	Prescale counter value	0

16.9.6 Match control register (MCR)

The match control register is used to control what operations are performed when one of the match registers matches the timer counter. Table 158 shows the function of each of the bits.

Table 158. MCR register (address 0x4000 4014) bit description

Bit	Symbol	Description		Reset value
0	MR0I	interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC		0
		0	disabled	
		1	enabled	
1	MR0R	reset on MR0: If MR0 matches the TC, then the TC is reset		0
		0	disabled	
		1	enabled	
2	MR0S	if MR0 matches the TC, the TC and PC are stopped and TCR[0] is set to logic 0.		0
		0	disabled	
		1	enabled	
3	MR1I	interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC.		0
		0	disabled	
		1	enabled	

Table 158. MCR register (address 0x4000 4014) bit description *...continued*

Bit	Symbol	Description		Reset value
4	MR1R	reset on MR1: If MR1 matches the TC, the TC is reset.		0
		0	disabled	
		1	enabled	
5	MR1S	stop on MR1: If MR1 matches the TC, the TC and PC are stopped and TCR[0] is set to logic 0.		0
		0	disabled	
		1	enabled	
6	MR2I	interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC.		0
		0	disabled	
		1	enabled	
7	MR2R	reset on MR2: If MR2 matches the TC, the TC is reset.		0
		0	disabled	
		1	enabled	
8	MR2S	stop on MR2: If MR2 matches the TC, the TC and PC are stopped and TCR[0] is set to logic 0.		0
		0	disabled	
		1	enabled	
9	MR3I	interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC.		0
		0	disabled	
		1	enabled	
10	MR3R	reset on MR3: If MR3 matches the TC, the TC is reset.		0
		0	disabled	
		1	enabled	
11	MR3S	stop on MR3: If MR3 matches the TC, the TC and PC are stopped and TCR[0] is set to logic 0.		0
		0	disabled	
		1	enabled	
31:12	-	reserved		-

16.9.7 Match registers (MR0/1/2/3)

The match register values are continuously compared to the timer counter value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the timer counter, or stop the timer. The settings in the MCR register control the actions.

Table 159. MR0/1/2/3 registers (addresses 0x4000 4018 to 24) bit description

Bit	Symbol	Description	Reset value
31:0	MATCH	timer counter match value	0

16.9.8 External match register (EMR)

The external match register provides control and status of the external match channels and external match pins CT32B_M[1:0].

If the match outputs are configured as PWM outputs in the PWMCON registers, PWM rules (see Section 16.9.10) determine the function of the external match registers.

Table 160. EMR register (address 0x4000 403C) bit description

Bit	Symbol	Description		Reset value
0	EM0	External match 0. This bit reflects the state of output CT32B_M0, regardless of this output being connected to its pin. When a match occurs between the TC and MR0, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[5:4] control the functionality of this output. If the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH), this bit is driven to the CT32B_M0 pins.		0
1	EM1	External match 1. This bit reflects the state of output CT32B_M1, regardless of this output being connected to its pin. When a match occurs between the TC and MR1, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[7:6] control the functionality of this output. This bit is driven to the CT32B_M1 pins.		0
3:2	-	reserved		-
5:4	EMC0	External match control 0. determines the functionality of external match 0.		0
		00b	do nothing.	
		01b	clear the corresponding external match bit/output to logic 0 (CT32B_M0 pin is LOW if pinned out).	
		10b	set the corresponding external match bit/output to logic 1 (CT326B_M0 pin is HIGH if pinned out).	
		11b	toggle the corresponding external match bit/output. ***Note:*** *The toggle only works when the corresponding match register is higher than 0.*	

Table 160. EMR register (address 0x4000 403C) bit description*...continued*

Bit	Symbol	Description		Reset value
7:6	EMC1	External match control 1. Determines the functionality of external match 1		0
		00b	do nothing	
		01b	clear the corresponding external match bit/output to logic 0 (CT32B_M1 pin is LOW if pinned out).	
		10b	set the corresponding external match bit/output to logic 1 (CT326B_M1 pin is HIGH if pinned out).	
		11b	toggle the corresponding external match bit/output. ***Note:*** *The toggle only works when the corresponding match register is higher than 0.*	

16.9.9 PWM control register (PWMC)

The PWM control register is used to configure the match outputs as PWM outputs. Each match output can be independently set as either a PWM output or match output. The external match register (EMR) controls the function of the match output.

Three single-edge controlled PWM outputs can be selected on the CT32B_M[1:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The match register that is configured to set the PWM cycle length, resets the timer. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Table 161. PWMC register (address 0x4000 4074) bit description

Bit	Symbol	Description		Reset value
0	PWMEN0	PWM channel0 enable		0
		0	EM0 controls the CT32B_M0	
		1	PWM mode is enabled for CT32B_M0	
1	PWMEN1	PWM channel1 enable		0
		0	EM1 controls the CT32B0_M1	
		1	PWM mode is enabled for CT32B_M1	
31:2	-	reserved		-

16.9.10 Rules for single edge controlled PWM outputs

1. All single-edge controlled PWM outputs go LOW at the beginning of each PWM cycle unless their match value is equal to zero.
2. When its match value is reached, each PWM output goes HIGH. If no match occurs (that is, the match value is greater than the PWM cycle length), the PWM output remains continuously LOW.
3. If a match value higher than the PWM cycle length is written to the match register and the PWM signal is HIGH already, the PWM signal is cleared on the start of the next PWM cycle.
4. If a match register value equals the timer reset value, the PWM output is reset to LOW on the next clock tick. So, the PWM output always consists of a one clock tick wide positive pulse. The PWM cycle length is the same as the timer reset value.
5. If a match register is set to zero, the PWM output goes HIGH the first time the timer goes back to zero and remains HIGH.

Note: *When the match outputs are used as PWM outputs, the reset and stop bits in the match control register (MCR) must be set to logic 0, except for the match register. The match register sets the PWM cycle length. For this register, set the MRR bit to logic 1 to enable the timer reset when the timer value matches the value of the corresponding match register.*

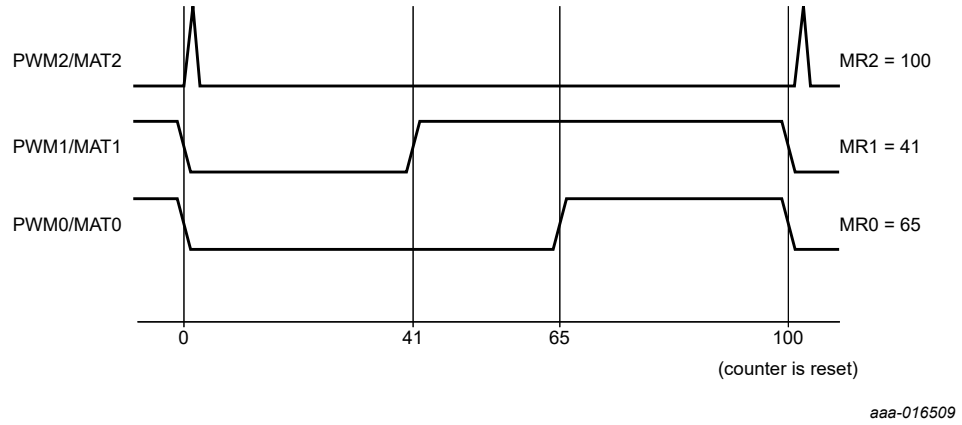


Figure 30. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR2) and MAT2:0 enabled as PWM outputs by the PWCM register

17 EEPROM controller

17.1 About this section

This section provides an overview of the EEPROM control and interface.

17.2 On-chip EEPROM

The NHS31xx contains a 4 kB EEPROM. This EEPROM is organized in 64 rows of 32 × 16-bit words. Of these rows, the last five contain calibration and test data and are locked. This data is either used by the bootloader after reset or made accessible to the application via firmware application programming interface (API).

17.3 Functional description

17.3.1 Reading from EEPROM

Reading is done via the AHB interface. The memory is mapped on the bus address space as a contiguous address space. Memory data words are seen on the bus using a little endian arrangement: Bits in the memory word are split in bytes so that byte 0 is made of bits 7:0, byte 1 by bits 15:8, and so on. Word 0, byte 0 is at byte address 0 in the bus, followed by word 0, byte 1, and so on, until word 0 is completed. The next address refers to word 1, byte 0, and so on.

The bus data width may have the same size as the memory data word, be wider or narrower.

If a read transfer request is detected, one or more memory read operations are performed sequentially until the required data is made available.

In the case of multiple memory read operations to satisfy a single bus read request, the order in which operations are performed is unspecified. So, all status bits which refer to the ’latest memory read’ are related to just one of the many read operations performed. To identify precisely which status applies to which word, use a read size which matches the memory data size. Conversely, flags refer to the full word even if only part of a memory word is read.

If the memory is busy with writes or other reads when a read request is placed on the bus, additional wait states are inserted until the requested read can be started. This action may be because, for instance, the memory is busy due to initialization or signature calculation.

If the memory is busy doing a program/erase when a read request is placed on the bus, an error response is given on the bus.

17.3.2 Writing to EEPROM

Erasing and programming is performed, as a single operation, on one or more words inside a single page.

Previous write operations have transferred the data to be programmed into the memory page buffer. The page buffer tracks which words were written to (offset within the page only). Words not written to, retain their previous content.

Note: *Do not write more than once on a position within a page register with different data, otherwise data loss may occur.*

When writing the binary value 110b inside the CMD register field, programming starts. The page specified in the last write operation is updated with the content of the page buffer, and the page buffer is cleared.

Note: *It is the act of writing 110b in CMD that starts programming, it is irrelevant if CMD already contained 110b from a previous programming operation.*

If another program/erase is in progress while the CMD register is written, the control interface returns an error. If any other memory operation is in progress, wait states are added until the memory is available.

The EEPAUTOPROG register allows an additional way of starting EEPROM programming:

- If EEPAUTOPROG contains the value 01b, EEPROM programming is also started after each successful write interface transaction.
- If EEPAUTOPROG contains the value 10b, EEPROM programming is started after each successful write interface transaction to the last memory word within a row. Writes to other memory words do not cause automatic programming.
- The EEPAUTOPROG value of 11b is reserved and must not be used.

17.4 Register descriptions

Table 162. Register overview: EEPROM memory controller (base address 0x4003 4000)

Name	Access	Address offset	Description	Reset value
CMD	R/W	0x000	EEPROM command register	0x00
RWSTATE	R/W	0x008	EEPROM read wait state register	-
PAUTOPROG	R/W	0x00C	EEPROM auto programming register	0x00
WSTATE	R/WL	0x010	EEPROM wait state register	-
CLKDIV	R/WL	0x014	EEPROM clock divider register	-
PWRDWN	R/W	0x018	EEPROM power-down register	0x00
MSSTART	R/W	0x020	EEPROM checksum start address register	-
MSSTOP	R/W	0x024	EEPROM checksum stop address register	-
MSDATASIG	R	0x028	EEPROM data signature register	-
MSPARSIG	R	0x02C	EEPROM parity signature register	-
STATUS	R	0x034	EEPROM device status register	-
MODULE_CONFIG	R	0xFD0	controller configuration options	-
INT_CLR_ENABLE	C	0xFD8	clear interrupt enable bits	0x00

Table 162. Register overview: EEPROM memory controller (base address 0x4003 4000) *...continued*

Name	Access	Address offset	Description	Reset value
INT_SET_ENABLE	S	0xFDC	set interrupt enable bits	0x00
INT_STATUS	R	0xFE0	interrupt status bits	0x00
INT_ENABLE	R	0xFE4	interrupt enable bits	0x00
INT_CLR_STATUS	C	0xFE8	clear interrupt status bits	-
INT_SET_STATUS	S	0xFEC	set interrupt status bits	-
MODULE_ID	R	0xFFC	controller memory module identification	0xEFCC 1101

17.4.1 Wait state registers (WSTATE, WSTATE)

The EEPROM controller has no awareness of absolute time, while for EEPROM operations several minimum absolute timing constraints have to be met. So it can only derive time from its clock by frequency division. The RWSTATE and WSTATE registers contain bit fields that specify the duration, in clock cycles, of diverse EEPROM operations. These fields are −1 encoded, so programming zero results in a duration of one clock cycle. For the values of the various phase durations, see the memory documentation.

Table 163. RWSTATE register (address 0x4003 4008) bit description

Bit	Symbol	Description	Reset value	Access
7:0	RPHASE2	duration of the EEPROM evaluation (read) phase	-	R/W
15:8	RPHASE1	duration of the EEPROM precharge phase	-	R/W
31:16	-	reserved	-	-

Table 164. WSTATE register (address 0x4003 4010) bit description

Bit	Symbol	Description	Reset value	Access
7:0	PHASE3	max. hold of all signals regarding write or program command	-	R/W
15:8	PHASE2	duration of the write or program command	-	
23:16	PHASE1	max. setup of any input regarding write or program command	-	R/W
30:24	-	reserved	-	-
31	LOCK_PARWEP	When set, all registers used for write/program timing configuration become read-only. So, logic 0 cannot be written to this bit after it has been set to logic 1.	0	

17.4.2 EEPROM auto-programming register (PAUTOPROG)

The auto-programming register allows the user to let the controller start an erase/program cycle automatically after a write interface access. It is not necessary to program the CMD register after data has been written.

Table 165. PAUTOPROG register (address 0x4003 400C) bit description

Bit	Symbol	Description		Reset value	Access
1:0	AUTOPROG	set auto-programming mode		0	R/W
		00b	auto-programming switched off		
		01b	erase/program cycle is triggered after 1 word is written		
		10b	erase/program cycle is triggered after a write to the last word of a page		
		11b	reserved		
31:2		reserved		-	R

17.4.3 EEPROM program clock management registers (CLKDIV, CLKDIV1, CLKDIV2)

The EEPROM requires a specific clock to manage program/erase cycles. The requirements for the frequency of this clock are given in the EEPROM specifications. The EEPROM controller generates this clock by dividing a timing reference clock by a programmable division factor. If configuration option ’reference_clock’ is selected, the timing reference clock is the dedicated input refclk. Otherwise, it is the system clock clk.

If the ’reference_clock’ option is selected, it is not possible to write to the clock management registers while program/erase is in progress.

Table 166. CLKDIV register (address 0x4003 4014) bit description

Bit	Symbol	Description	Reset value	Access
15:0	CLKDIV	0: program clock switched off; >0: clock division factor (-1 encoded)	0x0000	R
29:16	-	reserved	-	R
31:30	CLKMOD	amount of clock modulation (number of bits of random value added to CLKDIV at each clock cycle)	00b	-

Table 167. CLKDIV1 register (address 0x4003 4054) bit description

Bit	Symbol	Description	Reset value	Access
15:0	CLKDIV1	initial clock division factor, -1 encoded. Must always be CLKDIV	0x0000	R
31:16	CLKMOD	reserved	-	-

Table 168. CLKDIV2 register (address 0x4003 4058) bit description

Bit	Symbol	Description	Reset value	Access
15:0	CLKDIV2	number of program clock cycles after which the clock division factor is decremented, -1 encoded	0x0000	R
31:16	-	reserved	-	-

17.4.4 EEPROM power down register (PWRDWN)

This register enables the EEPROM to enter power-down.

After exiting power-down, it is necessary to wait for a specific power-up time (100 µs), before operation is resumed. The controller does not check for these conditions, nor does it block illegal operations.

The controller does not track if power-down is in effect. It does not prevent its activation when the memory is busy. So it is forbidden to activate power-down while any memory operation is in progress. It is also forbidden to initiate any operation when the memory is in power-down or recovering from power-down. In relation to the above statements, initialization (which follows exit from reset) is considered a memory operation.

After exiting power-down, it is necessary to wait for a specific power-up time (100 µs), before operation is resumed. The controller does not check for these conditions, nor does it block illegal operations. The VMPOK bit in the EEPROM status register indicates that the EEPROM is ready.

Table 169. PWRDWN register (address 0x4003 4020) bit description

Bit	Symbol	Description	Reset value	Access
0	PWRDWN	when bit is set, the power-down mode is in effect	0	R
31:1	-	reserved	-	-

17.4.5 Signature generation registers

The controller is able to compute a memory content signature across a user-specified address range. A data signature and a parity signature are computed. The data signature is based on ECC-corrected data, while the parity signature is based on uncorrected data.

Table 170. MSSTART register (address 0x4003 4020) bit description

Bit	Symbol	Description	Reset value	Access
15:0	STARTA	when bit is set, power-down mode is in effect	0x0000	R
31:16	-	reserved	-	-

Table 171. MSSTOP register (address 0x4003 4024) bit description

Bit	Symbol	Description	Reset value	Access
15:0	STOPA	stop address for signature generation (the word specified by STOPA is included in the address range). The address is in units of memory words, not bytes. Bits 2:0 cannot be written and are forced to 111.	0x0000	R/W

Table 171. MSSTOP register (address 0x4003 4024) bit description*...continued*

Bit	Symbol	Description	Reset value	Access
30:16	-	reserved	-	R
31	STRTBIST	when this bit is written to 1, signature generation starts. At the end of signature generation, this bit is automatically cleared.	0	R/W

Table 172. MSDATASIG register (address 0x4003 4028) bit description

Bit	Symbol	Description	Reset value	Access
31:0	DATA_SIG	after signature generation has run, this field contains the data signature for the specified address range.	0x0000 0000	W

Table 173. MSPARSIG register (address 0x4003 402C) bit description

Bit	Symbol	Description	Reset value	Access
0	PARITY_SIG	after signature generation has run, this field contains the parity signature for the specified address range.	0	W
31:1	-	reserved	-	-

17.4.6 Registers with status information

Table 174. STATUS register (address 0x4003 4034) bit description

Bit	Symbol	Description	Reset value	Access
3:0	HVTRIM_E	when bit is set, power-down mode is in effect	0	R
7:4	HVTRIM_P	trim value for the programming voltage	-	R
8	CORRECTED	ECC detected a correctable error	-	R
9	INVALID	ECC detected an uncorrectable error	-	R
10	ALL1	all bits in the last memory word read are high	-	R
11	ALL0	all bits in the last memory word read are Low	-	R
12	HVERR	high-voltage error	-	R
13	TMANALOG	Analog test mode is in effect	-	R
14	VMPOK	margin pump level OK	-	R
31:15	-	reserved	-	R

17.4.7 Interrupt registers

These registers determine when the EEPROM controller gives an interrupt request. When the bit-wise AND of INT_STATUS and INT_ENABLE is non-zero, the int_req output is asserted.

If the corresponding INT_ENABLE bit is logic 0, an INT_STATUS register bit can be polled to test for the occurrence of an event.

The INT_STATUS register can be set for software testing purpose by writing into the INT_SET_STATUS register.

Table 175. INT_CLR_ENABLE register (address 0x4003 4FD8) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CLR_ENABLE	When the corresponding CLR_ENABLE bit set to logic 1, the INT_ENABLE bit is cleared.	0x0000 0000	-

Table 176. INT_SET_ENABLE register (address 0x4003 4FDC) bit description

Bit	Symbol	Description	Reset value	Access
31:0	SET_ENABLE	when a SET_ENABLE bit is set to logic 1, the corresponding INT_ ENABLE bit is set, unless it is a reserved bit.	0x0000 0000	-

Table 177. INT_CLR_STATUS register (address 0x4003 4FE8) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CLR_STATUS	when a CLR_STATUS bit is set to logic 1, the corresponding INT_ STATUS bit is cleared.	0x0000 0000	-

Table 178. INT_ENABLE register (address 0x4003 4FE0) bit description

Bit	Symbol	Description	Reset value	Access
31:0	INT_ENABLE	if an INT_ENABLE bit is set to logic 1, an interrupt request is generated if the corresponding INT_STATUS bit is asserted.	0x0000 0000	-

Table 179. INT_STATUS register (address 0x4003 4FE4) bit description

Bit	Symbol	Description	Reset value	Access
0	-	reserved	0	-
1	SIG_DONE	this status bit is set at the end of signature computation.	0	
2	PROG_DONE	this status bit is set at the end of a high-voltage operation (burn, erase, combined erase+program).	0	
3	ECC_ERR	this status bit is set if, during a memory read operation, the ECC decoding logic detects a correctable or uncorrectable error. Either a user-requested read, a speculative read, or reads performed during initialization or during signature generation could set this bit.	0	
31:4	-	reserved	-	

17.4.8 Module ID register (MODULE_ID)

Table 180. MODULE_ID register (address 0x4003 4FFC) bit description

Bit	Symbol	Description	Reset value	Access
7:0	APERTURE	the size of the memory encoded as (size/4kB) − 1	0x01	R
11:8	MINOR_REV	minor revision of module implementation	0x01	R
15:12	MAJOR_REV	major revision of module implementation	0x01	R
31:16	-	identification number; unique module identifier indicating the module ID of the controller	0xEFCC	R

18 Flash controller

18.1 About this chapter

This chapter describes how to read and write to flash by accessing the flash controller registers directly. To support flash erasing and writing, functions are included in the IC. These in-application-programming (IAP) functions are described in Section 6.

18.2 On-chip flash

The NHS31xx family contain a 32 kB flash memory organized in 32 sectors of 1 kB each. Each sector consists of 16 rows of 16 × 32-bit words. The two topmost sectors contain the initialization code and IAP firmware.

Note: *During an erase or program operation, the flash is not accessible.*

18.3 Features

The Flash controller provides control mechanisms for the flash memory. A detailed description can be found in the NVMC IP data sheet.

18.4 Functional description

18.4.1 Reading from flash

Reading is done via the AHB interface. The memory is mapped on the bus address space as a contiguous address space. Memory data words are seen on the bus using a little endian arrangement: Bits in the memory word are grouped in bytes so that byte 0 is made of bits 7:0, byte 1 by bits 15:8, and so on. Word 0, byte 0 is at byte address 0 in the bus, followed by word 0, byte 1, and so on, until word 0 is completed. The next address refers to word 1, byte 0, and so on.

The bus data width may have the same size as the memory data word, be wider or narrower.

If a read transfer request is detected, one or more memory read operations are performed sequentially until the required data is made available.

In the case of multiple memory read operations to satisfy a single bus read request, the order in which operations are performed is unspecified. So, all status bits which refer to the ’latest memory read’ are related to just one of the many read operations performed. To identify precisely which status applies to which word, use a read size which matches the memory data size. Conversely, flags refer to the full word even if only part of a memory word is read.

If, when a read request is placed on the bus, the memory is busy, additional wait states are inserted until the requested read can be started. This action may be because, for instance, the memory is busy due to writes, other reads, initialization, or signature calculation.

If, when a read request is placed on the bus, the memory is busy doing program/erase, an error response is given on the bus. This request affects instruction fetch operations from the Arm core. So, program code must be moved to SRAM before starting an erase/program cycle.

18.4.2 Writing to flash

Writing to flash means copying a word of data over the AHB to the page buffer of the flash. It does not actually program the data in the memory array. Subsequent erase and program cycles do this programming.

On flash, there are two operations (sector selection and protection/unprotection) which require a variant of the ’standard’ write cycle described below. To distinguish between these three functions, setting FCTR register bits as follows before a write cycle is required:

- For page-register write; WPB=0, WEB=1, WRE=1, CS=1
- For sector selection; WPB=1, WEB=1, WRE=0, CS=1
- For sector protection/unprotection; WPB=1, WEB=1, WRE=1, CS=1

Note: *Reading is not possible in all of the above cases.*

In general, the row/sector address is not significant during writing itself, since writing only accesses the page buffer, which is one row wide. Nonetheless, during writing the full address is used in the following ways:

- Address range checks
- The address of the last written location can be used to identify the row to be programmed. This usage can be done in case of automatic programming after a write and if a subsequent programming is triggered through the control interface.

If the memory is idle, posted writes are performed. The bus transaction completes immediately and then the controller moves data to the memory while the bus is free for other non-memory-related transactions. However, if the memory is busy doing reads or previous writes, additional wait states are inserted until the memory is free. The bus write is then completed and the controller-to-memory data transfer is started.

If, when a write request is placed on the bus, the memory is busy doing program/erase, an error response is given on the bus.

After a write, the controller retains the last data written and the last address (possibly incremented, in case of the ’register’ write interface). It is possible that write address and data were provided previously, but the actual write cycle was not performed because the memory was not configured.

It is possible to start a memory write cycle using the previously provided information. This cycle can be done by setting the LOADREQ bit in the FCTR register. Other FCTR bits are configured to enable writing, sector selection, or protection/unprotection. Bit LOADREQ always reads as logic 0.

18.4.3 Erasing/programming flash

Erasing and programming are separate operations. Both are possible only on memory sectors that are unprotected and unlocked. Protect/lock information is stored inside the memory itself, so the controller is not aware of protection status. So, if a program/erase operation is performed on a protected or locked sector, it does not flag an error.

Locking - Each flash sector has a lock bit which is stored in the memory itself and loaded on exit from reset from the index sector. The user cannot modify this lock bit. Only its reset value in the index sector can be modified. Sectors containing firmware are locked by default.

Protection - At exit from reset, all sectors are protected against accidental modification. To allow modification, a sector must be unprotected. It can then be protected again after that the modification is performed.

To protect/unprotect a sector, the following operations must be performed:

1. The FCTR register must be configured with bits WPB=1, WEB=1, WRE=1, CS=1. Reading is not possible in this case.
2. A memory write must be performed inside the address range of the sector to be protected/unprotected. Bit 0 of the memory word written updates the protection status (logic 0 = unprotected, logic 1 = protected).

Erasing can be performed on one or more sectors. If more than one sector must be erased, additional sectors must be selected first.

Erasing sets the content of all bits in the selected memory zone to logic 1.

To select a sector for erasing, the following operations must be performed:

1. The FCTR register must be configured with bits WPB=1, WEB=1, WRE=0, CS=1. Reading is not possible in this case.
2. A memory write must be performed inside the address range of the sector to be selected.

When a sector is selected, there is no means to deselect it before erase (it is deselected automatically at the end of erase). If a sector is mistakenly selected, it can be protected and an erase cycle started, which clears the selection latches. In this case, no sectors are erased.

Erase is also performed on the sector containing the address to which the latest write command was given prior to the erase command. In case of a page erase, only the page containing that address is erased.

To start erase, the following operations must be performed:

1. The TR field in the FPTR register must be initialized with the required erase pulse duration and the EN_T bit must be set.
2. The bit combination WPB=1, WEB=0, WRE=0, CS=1 must be written in the FCTR register. In case of page erase, ERSP must be set to logic 1, otherwise ERSP must be logic 0.

During erase, the RY bit in the FSTAT register is logic 0. It reverts to logic 1 at erase end.

At the end of erase, the PROG_DONE bit in the INT_STATUS register is set and an interrupt is requested, if enabled. FCTR bit WEB is set and ERSP is cleared.

Programming can be performed on a flash page. During programming, bits which are logic 0 in the page register are set to logic 0 also in the selected memory page. Bits which are logic 1 in the page register remain unchanged in the memory page.

Due to the presence of ECC, it is not allowed to modify additional bits inside a memory word where some bits have already been programmed. If allowed, the resulting ECC code in memory is then the AND of the codes for the previous and new values written. This value can be inconsistent with the resulting data, potentially resulting in unwanted or missing bit corrections, or spurious error conditions.

If bit PDL in the FCTR register is set, the page register is set to all logic 1.

PDL must be manually cleared before writing the page register. Then the page register must be written with the data to be programmed, before starting the programming operation.

The page addressed in the latest write operation is programmed. To start programming, the following operations must be performed:

1. The TR field in the FPTR register must be initialized with the required program pulse duration (number of timing reference clock cycles: either clk or refclk clock is used, depending on the 'reference_clock' configuration option) and the EN_T bit must be set.
2. The bit combination WPB=1, WEB=0, WRE=1, CS=1 must be written in the FCTR register. ERSP must be 0.

During programming, the RY bit in the FSTAT register is logic 0. It reverts to logic 1 at end of program.

At the end of program, the PROG_DONE bit in the INT_STATUS register is set and an interrupt is requested, if enabled. FCTR bit WEB is set.

Note: *During an erase or program operation, the flash is not accessible. Hence, if the interrupt vector table is placed in flash (default setup) the application ensures that no interrupts are triggered. Accessing the table, if in flash, results in a hard fault.*

18.4.4 Signature computation

This function reads all memory words within a specified address range, computing signatures for the read data and associated parity bits. Software can then compare the two signatures with the corresponding expected values: if both data and parity signatures match, then the memory content is as expected. If only the data signature matches, some memory bits were wrong, but ECC managed to correct these errors, so user data is still correct. If the data signature is wrong, the memory content does not match the expected data.

The address range is specified by writing the start address in register field STARTA, and the end address in register field STOPA. Both of these addresses are included in the range. These addresses are memory word addresses, not bus addresses, i.e. they represent the byte address divided by the size in bytes of a memory word. It is not possible to specify an address in the index sector address space (i.e. with the MSB set). However, it is possible to set the ISS bit in the FCTR register and thus perform a check on the index sector content.

Setting register bit STRTBIST starts signature computation. Since STOPA and STRTBIST are fields of the same register, they can be set with a single control write operation. At the end of the signature computation, computed data and parity signatures are available in the DATA_SIG and PARITY_SIG register fields.

The SIGNATURE_DONE interrupt status bit is set at the end of signature computation.

While the controller is busy doing signature computation, all reads and writes, both on the memory interfaces and on the control interface, are stalled.

Both data and parity signatures are computed through MISR registers, which are zeroed at the start of signature computation. They are updated in the following way every time a new data word is made available:

- XOR is applied to some selected bits (shown in Table 181) of the previous signature value and the result is placed in bit 0
- The resulting value is rotated right 1 bit (i.e. bit 1 goes into bit 0, bit 0 goes into the MSB, and so on,…)
- XOR is applied to the input data, bit by bit, with the result of the above mentioned rotation

- The result of this last operation is the new signature value

Table 181. Feedback bits for MISR computation

signature size	1st tap position	2nd tap position	3rd tap position
16	4	13	15
32	10	30	31
128	2	27	29

For the data signature, read data (after a possible ECC correction) is the input data of the data signature. Thus, the DATA_SIG register has the same size as the memory word. An exception is when the memory data word is 256 bits wide, the signature is still 128 bits wide: in this case, data is sent to the MISR twice, first bits 127-0 and then bits 255-128, applying each time the algorithm described above.

To reduce the possibility that invalid content has the same signature as correct content, the parity signature is always 16 bits wide, even though parity bits are less. Input data for the signature algorithm is obtained as follows: uncorrected parity bits occupy the least significant bits, followed by the ECC 'CORRECTED' flag and then the ECC 'INVALID' error flag. Finally, zeros are used to extend the value to 16. In analogy with the data signature, for 256-bit memories, the parity signature is also updated twice for each word. It is updated using the 8 parity bits and the two ECC flags pertaining to each half of the 256-bit word, in the same order.

18.5 Register descriptions

Table 182. Register overview: Flash memory controller (base address 0x4004 C000)

Name	Access	Address offset	Description	Reset value
FCTR	R/W	0x000	flash control register	0x5
FSTAT	R	0x004	flash status register	-
FPTR	R/W	0x008	flash program-time register	0x0
FBWST	R/W	0x010	flash wait state register	
FCRA	R/W	0x01C	flash program clock divider	0x0
FMSSTART	R/W	0x020	flash checksum start address register	0x0
FMSSTOP	R/W	0x024	flash checksum stop address register	0x0
FMS16	R	0x028	flash parity signature register	-
FMSW0	R	0x02C	flash data signature register	-
ECCRSTERRCNT	R	0x050	ECC status information	0x0
ECCERRCNT	R/W	0x054	invalid flag and error corrected counter reset	0x0
MODULE_CONFIG	R	0xFD0	controller configuration options	
INT_CLR_ENABLE	C	0xFD8	clear interrupt enable bits	0x0
INT_SET_ENABLE	S	0xFDC	set interrupt enable bits	0x0
INT_STATUS	R	0xFE0	interrupt status bits	0x0
INT_ENABLE	R	0xFE4	interrupt enable bits	0x0
INT_CLR_STATUS	C	0xFE8	clear interrupt status bits	

Table 182. Register overview: Flash memory controller (base address 0x4004 C000)*...continued*

Name	Access	Address offset	Description	Reset value
INT_SET_STATUS	S	0xFEC	set interrupt status bits	
MODULE_ID	R	0xFFC	controller memory module identification	0xEFCC 110F

18.5.1 Flash control register (FCTR)

The flash control register is used to select read modes and to control the programming of the flash memory.

Table 183. FCTR register (address 0x4003 C000) bit description

Bit	Symbol	Description	Reset value	Access
0	CS	when logic 0, the device is in standby mode. CS must be logic 1 for proper operation.	0x1	R/W
1	WRE	when bit WPB=1, this bit selects between erase (WRE=0) and program (WRE=1). When WPB=0, it selects between read (WRE=0) and write (WRE=1).	0x0	R/W
2	WEB	when this bit is cleared, program/erase starts. It is automatically set to logic 1 at the end of the operation.	0x1	R/W
3	-	reserved	-	-
4	DCR	when logic 1, DC read mode is selected.	0x0	R/W
5	RLD	when logic 1, the page register data latches are read instead of the matrix. Row and sector address bits are not significant.	0x0	R/W
6	ISS	when logic 1, the index sector is accessed instead of the main array. When logic 0, the MSB of the address bus determines index sector access.	0x0	R/W
7	WPB	when logic 1, program/erase is enabled. When 0, read or write is selected, depending on bit WEB.	0	R/W
8	ERSP	if set to logic 1, single page erase mode is enabled. The bit is automatically reset at the end of the erase cycle.	0	R/W
9	PD	when high, the memory is forced to enter the power-down mode.	0	R/W
10	PDL	when high, the page register is preset to all logic 1.	0	R/W
11	RLS	when logic 1, sector latches are read instead of the memory array.	0	R/W
12	PROGREQ	logic 1 whenever the memory is busy doing program/erase.	0	R
13	-	reserved	-	-
14	-	reserved	-	-
15	LOADREQ	When it is written with logic 1, a memory write operation is triggered. Always reads as logic 0.	0	R/W
16	REMAP	when logic 1, address remapping is enabled.	0	R/W
17	INIT_WR	writing a logic 1 to this field together with reset values for all other fields of the FCTR register starts write initialization. When set, only a controller reset clears this bit.	0	R/W
18	LPM	when set, the memory enters the low-power mode	1	R/W
19	PDBG	when set, band gap enters the power-down mode	1	R/W

Table 183. FCTR register (address 0x4003 C000) bit description*...continued*

Bit	Symbol	Description	Reset value	Access
31:20	-	reserved	-	-

The flash initialization has two stages: the read mode initialization is started after reset. The write mode initialization is triggered (once) by writing logic 1 to the INIT_WR bit. Subsequent writes are ignored. Flash access during an initialization cycle results in a bus stall.

18.5.2 Flash status register (FSTAT)

Table 184. FSTAT register (address 0x4003 C010) bit description

Bit	Symbol	Description	Reset value	Access
1:0	-	reserved	0	R
2	RY	RY = logic 1 whenever the flash is not busy reading, programming or erasing.	0	R
4:3		reserved	-	-
7:5	SL	sector latches: bit 2 = selected, 1 = locked, 0 = protected.	0	R
8	CORRECTED	ECC detected a correctable error.	0	R
31:9	-	reserved	-	-

18.5.3 Flash wait state register (FBWST)

To manage memory operation, it is often necessary to wait for some time. The only time reference that the controller has, is its clock. So, all time delays have to be expressed in terms of number of clock cycles.

The reference value for all delays is in the data sheet of the corresponding memory. To specify a value to the controller, the delay value quoted in the data sheet must be divided by the period of the clk clock. A non-integer result must be rounded up, since delays are normally minimum delays.

To specify delay values, the controller uses ’−1-based encoding’ inside register bit fields. The number of clock cycles minus 1 is written to these fields. So each delay has a minimum value of 1 clk cycle, corresponding to the code logic 0.

Table 185. FBWST register (address 0x4003 C010) bit description

Bit	Symbol	Description	Reset value	Access
7:0	WST	number of wait states	0x00	R/W
13:8	-	reserved	0x0	R
14	SPECALWAYS	speculative reading: logic 0 - single speculative reading; logic 1 - always speculative reading.	1	R/W
15	CACHE2EN	when caching is enabled: if this bit is logic 1, more than a word can be cached; otherwise only the last word read is kept in the cache. When caching is disabled or unavailable, this bit has no effect.	1	R/W
31:8	-	reserved	-	R

Table 186. WST values for different Flash operating modes (FCTR settings) and system clock speed

System clock	Read/write mode	Low-power Read mode
8 MHz	0x0000 0000	0x0000 0001
4 MHz /62.5 kHz	0x0000 0000	0x0000 0000

18.5.4 Registers for program/erase management

The following registers provide timing references required during program and erase operations. FPTR selects the duration of the program/erase operation and FCRA selects the frequency of the program/erase clock.

Table 187. FPTR register (address 0x4003 C008) bit description

Bit	Symbol	Description	Reset value	Access
11:0	EN_T	when set, timer starts. It is automatically cleared when TR reaches logic 0.	0x000	R
28:11	TR	duration of the program/erase pulse, in timing reference clock cycles.	0x0	R
31:29	-	reserved	-	-

Table 188. FCRA register (address 0x4003 C01C) bit description

Bit	Symbol	Description	Reset value	Access
11:0	FCRA	when logic 0, FCRA clock is stopped. Otherwise, its frequency is the timing reference clock frequency divided by (FCRA + 1).	0x000	R/W
31:12	-	reserved	-	-

If the reference_clock option is selected, when the program/erase clock is started, register FCRA can only be written with logic 0 to stop the program/erase clock. Writing any other value reports an error. It is also not possible to stop the clock during program or erase. Furthermore, after writing FCRA, it is necessary to wait 2 clk clock cycles and 2 refclk clock cycles before the register is writable again.

18.5.5 Signature generation registers

The controller can compute a memory-content signature across a user-specified address range. A data signature and a parity signature are computed. The data signature is based on ECC-corrected data, while the parity signature is based on uncorrected data.

Table 189. FMSSTART register (address 0x4003 C020) bit description

Bit	Symbol	Description	Reset value	Access
16:0	STARTA	start address for signature generation. The address is in units of memory words, not bytes. Bits 2:0 cannot be written and are forced to 000b.	0	R/W
31:17	-	reserved	-	-

Table 190. FMSSTOP register (address 0x4003 C024) bit description
[1]

Bit	Symbol	Description	Reset value	Access
16:0	STOPA	stop address for signature generation (the word specified by STOPA is included in the address range). The address is in units of memory words, not bytes. Bits 2:0 cannot be written and are forced to 111b.	0	R/W
30:17	-	reserved	-	-
31	STRTBIST	when this bit is written as logic 1, signature generation starts. At the end of signature generation, this bit is automatically cleared.	0	-

[1] Index sector addresses cannot be specified in FMSSTART and FMSSTOP.

Table 191. FMS16 register (address 0x4003 C028) bit description

Bit	Symbol	Description	Reset value	Access
15:0	-	after signature generation has run, this field contains the parity signature for the specified address range.	0x0000	R/W
31:16	-	reserved	-	-

Table 192. FMSW0 register (address 0x4003 C02C) bit description

Bit	Symbol	Description	Reset value	Access
31:0	-	after signature generation has run, this field contains the data signature for the specified address range.	0x0000 0000	R/W
31:16	-	reserved	-	-

18.5.6 Registers with ECC status information

Table 193. ECCRSTERRCNT register (address 0x4003 C050) bit description

Bit	Symbol	Description	Reset value	Access
0	ECCRSTERRCNT	when set to logic 1, it clears the ECCERRCNT register. Always reads as logic 0.	0	R
31:1	-	reserved	-	-

Table 194. ECCERRCNT register (address 0x4003 C054) bit description

Bit	Symbol	Description	Reset value	Access
0	INVALID_FLAG	set when an uncorrectable ECC error occurs. Cleared by writing logic 1 into ECCRSTERRCNT.	0	R
4:1	ERRCOUNTER	incremented when an ECC correction occurs (saturates at 1111b). Cleared by writing logic 1 into ECCRSTERRCNT.		
17:5	ERR_PAGE	page address of the last Location containing an ECC (correctable or uncorrectable) error. Only significant if ERRCOUNTER or INVALID_ FLAG are non-zero.		
31:18	-	reserved	-	-

18.5.7 Interrupt registers

These registers determine when the NVMC gives an interrupt request. When the bit-wise AND of INT_STATUS and INT_ENABLE is non-zero, the int_req output is asserted.

If the corresponding INT_ENABLE bit is logic 0, an INT_STATUS register bit can be polled to test for the occurrence of an event.

The INT_STATUS register can be set for software testing purpose, by writing into the INT_SET_STATUS register.

Table 195. INT_CLR_ENABLE register (address 0x4003 CFD8) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CLR_ENABLE	when a CLR_ENABLE bit is set to logic 1, the corresponding INT_ ENABLE bit is cleared.	0x0000 0000	-

Table 196. INT_SET_ENABLE register (address 0x4003 CFDC) bit description

Bit	Symbol	Description	Reset value	Access
31:0	SET_ENABLE	when a SET_ENABLE bit is set to logic 1, the corresponding INT_ ENABLE bit is set, unless it is a reserved bit.	0x0000 0000	-

Table 197. INT_CLR_STATUS register (address 0x4003 CFE8) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CLR_STATUS	when a CLR_STATUS bit is set to logic 1, the corresponding INT_ STATUS bit is cleared.	0x0000 0000	-

Table 198. INT_SET_STATUS register (address 0x4003 CFEC) bit description

Bit	Symbol	Description	Reset value	Access
31:0	SET_STATUS	when a SET_STATUS bit is set to logic 1, the corresponding INT_ STATUS bit is set, unless it is a reserved bit.	0x0000 0000	-

Table 199. INT_ENABLE register (address 0x4003 CFE0) bit description

Bit	Symbol	Description	Reset value	Access
31:0	INT_ENABLE	if an INT_ENABLE bit is set to logic 1, an interrupt request is generated of the corresponding INT_STATUS bit is asserted.	0x0000 0000	-

Table 200. INT_STATUS register (address 0x4003 CFE4) bit description

Bit	Symbol	Description	Reset value	Access
0	-	reserved	-	-
1	SIG_DONE	this status bit is set to logic 1 at the end of signature computation.	0	-
2	PROG_DONE	this status bit is set to logic 1 at the end of a high-voltage operation (burn, erase, combined erase+program).	0	-
3	ECC_ERR	this status bit is set to logic 1 if, during a memory read operation, ECC decoding logic detects a correctable or uncorrectable error. This read may be either a user-requested read, a speculative read, or reads performed during initialization or during signature generation.	0	-
31:4	-	reserved	-	-

18.5.8 Module ID register (MODULE_ID)

Table 201. MODULE_ID register (address 0x4003 CFFC) bit description

Bit	Symbol	Description	Reset value	Access
7:0	APERTURE	the size of the memory encoded as (size / 4 kB) − 1.	0x0F	R
11:8	MINOR_REV	minor revision of module implementation	0x1	R
15:12	CORRECTED	major revision of module implementation	0x1	R
31:16	-	identification number; unique module identifier indicating the module ID of the controller.	0xEFCC	R

19 I²C-bus interface

19.1 About this chapter

This chapter describes the I²C-bus operation and the software interface. It also explains how to use the I²C-bus for wake-up from reduced power modes.

The I²C-bus interface is available on all NHS31xx family members.

19.2 General description

Figure 31 shows a typical I²C-bus configuration. Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I²C-bus:

- Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. A number of data bytes follows. The slave returns an acknowledge (ACK) bit after each received byte.
- Data transfer from a slave transmitter to a master receiver. The master transmits the first byte (the slave address). The slave responds by returning an ACK bit. The slave then transmits the data bytes to the master. The master returns an ACK bit after all received bytes other than the last byte. At the end of the last received byte, a 'not-acknowledge' (NACK) is returned. The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated-START condition. As a Repeated-START condition is also the beginning of the next serial transfer, the I²C bus is not released.

The I²C-bus interface is byte oriented and has four operating modes: master transmitter mode, master receiver mode, slave transmitter mode, and slave receiver mode.

The I²C interface is completely I²C-bus compliant, supporting the ability to power off the NHS31xx independent of other devices on the same I²C-bus.

The I²C-bus interface receives an 8 MHz clock signal from the system oscillator.

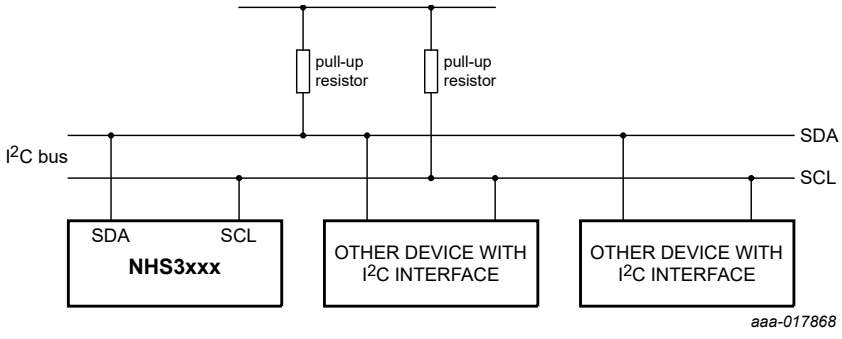


Figure 31. Typical I²C-bus configuration

19.3 Features

Standard I^2C-bus compliant bus interfaces may be configured as master, slave, or master/slave.

- Arbitration is handled between simultaneously transmitting masters without corruption of serial data on the bus.
- Programmable clock allows adjustment of I^2C-bus transfer rates.
- Data transfer is bidirectional between masters and slaves.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization is used as a handshake mechanism to suspend and resume serial transfer.
- Supports standard mode (100 kbit/s) and fast mode (400 kbit/s)
- Optional recognition of up to four slave addresses
- Monitor mode allows observing all I^2C-bus traffic, regardless of slave address.
- The I^2C-bus can be used for test and diagnostic purposes.
- The I^2C-bus contains a standard I^2C-bus-compliant bus interface with two pins.
- Possible to wake up NHS31xx on matching I^2C-bus slave address

19.4 Configuration

The I^2C-bus interface is configured using the registers in the APB block at address 0x4000 0000, plus the following:

- The I^2C-bus pin functions and the I^2C-bus mode are configured in the IOCON register block.
- Power and peripheral clock: In the SYSAHBCLKCTRL register, set bit 5.
- Reset
  Before accessing the I^2C-bus block, ensure that the I2C_RST_N bit (bit 1) in the PRESETCTRL register is set to logic 1. Setting this bit deasserts the reset signal to the I^2C-bus block.

19.5 Applications

Interfaces to external I^2C-bus standard parts, such as serial RAMs, LCDs, tone generators, and other microcontrollers.

19.6 Pin description

The I^2C-bus pins must be configured through the PIO0_4 (see Table 259) and PIO0_5 (see Table 260) registers for standard mode or fast mode. The I^2C-bus pins are open-drain outputs and fully compatible with the I^2C-bus specification.

Table 202. I^2C-bus pin description

Pin	Type	Description
SDA	I/O	I^2C-bus serial data
SCL	I/O	I^2C-bus serial Clock

19.7 Register descriptions

Table 203 shows the register addresses of the I²C-bus controller.

The reset values reflect the data stored in used bits only. It does not include the content of reserved bits.

Table 203. Register overview of the I²C-bus controller (base 0x4000 000)

Name	Access	Address offset	Description	Reset value
CONSET	R/W	0x000	I²C-bus control set register. When a logic 1 is written to a bit of this register, the corresponding bit in the I²C-bus control register is set. Writing a logic 0 has no effect on the corresponding bit in the I²C-bus control register.	0
STAT	R	0x004	I²C-bus status register. During I²C-bus operation, this register provides detailed status codes that allow software to determine the next action needed.	0xF8
DAT	R/W	0x008	I²C-bus data register. During master or slave transmit mode, data to be transmitted is written to this register. During master receive mode or slave receive mode, data that has been received may be read from this register.	0
ADR0	R/W	0x00C	I²C-bus slave address register 0. Contains the 7-bit slave address for operation of the I²C-bus interface in slave mode and is not used in master mode. The least significant bit determines if a slave responds to the general call address. This address is also used for device wake-up.	0
SCLH	R/W	0x010	SCH duty cycle register high halfword. Determines the high time of the I²C-bus clock.	0
SCLL	R/W	0x014	SCL duty cycle register Low halfword. Determines the low time of the I²C-bus clock. I2nSCLL and I2nSCLH together determine the clock frequency generated by an I²C-bus master and certain times used in slave mode.	0
CONCLR	R/W	0x018	I²C-bus control clear register. When a logic 1 is written to a bit of this register, the corresponding bit in the I²C-bus control register is cleared. Writing a logic 0 has no effect on the corresponding bit in the I²C-bus control register.	0
MMCTRL	R/W	0x01C	Monitor mode control register	0
ADR1	R/W	0x020	I²C-bus slave address register 1. Contains the 7-bit slave address for operation of the I²C-bus interface in slave mode. It is not used in master mode. The least significant bit determines if a slave responds to the general call address.	0
ADR2	R/W	0x024	I²C-bus slave address register 2. Contains the 7-bit slave address for operation of the I²C-bus interface in slave mode. It is not used in master mode. The least significant bit determines if a slave responds to the general call address.	0
ADR3	R/W	0x028	I²C-bus slave address register 3. Contains the 7-bit slave address for operation of the I²C-bus interface in slave mode. It is not used in master mode. The least significant bit determines if a slave responds to the general call address.	0

Table 203. Register overview of the I²C-bus controller (base 0x4000 000) *...continued*

Name	Access	Address offset	Description	Reset value
DATA_BUFFER	R	0x02C	Data buffer register. After every 9 bits have been received on the bus, the 8 MSBs of the I2DAT shift register are automatically transferred to the DATA_BUFFER. These 9 bits consist of 8 data bits plus ACK or NACK.	0
MASK0	R/W	0x030	I²C-bus slave address mask register 0. This mask register is associated with I2ADR0 to determine an address match. The mask register has no effect when comparing to the general call address (‘000 0000’).	0
MASK1	R/W	0x034	I²C-bus slave address mask register 0. To determine an address match, this mask register is associated with I2ADR0. The mask register has no effect when comparing to the general call address (‘000 0000’).	0
MASK2	R/W	0x038	I²C-bus slave address mask register 0. To determine an address match, this mask register is associated with I2ADR0. The mask register has no effect when comparing to the general call address (‘000 0000’).	0
MASK3	R/W	0x03C	I²C-bus slave address mask register 0. To determine an address match, this mask register is associated with I2ADR0. The mask register has no effect when comparing to the general call address (‘000 0000’).	0

19.7.1 I²C control set register (CONSET)

The CONSET register manages bits in the CON register that control the operation of the I²C-bus interface. Writing a logic 1 to a bit of this register causes the corresponding bit in the I²C-bus control register to be set. Writing a logic 0 has no effect.

Table 204. CONSET register (address 0x4000 0000) bit description

Bit	Symbol	Description	Reset value
1:0	-	reserved	0
2	AA	Assert ACK flag	0
3	SI	I²C-bus interrupt flag	0
4	STO	STOP flag	0
5	STA	START flag	0
6	I2EN	I²C-bus interface enable	0
31:7	-	reserved	0

AA

Assert acknowledge Flag. When set to logic 1, an acknowledge (LOW level to SDA) is returned during the acknowledge clock pulse on the SCL line on the following situations:

- The address in the slave address register has been received.
- The general call address has been received while the general call bit (GC) in the ADR register is set.
- A data byte has been received while the I²C-bus is in the master receiver mode.

- A data byte has been received while the I^2C-bus is in the addressed slave receiver mode.

Writing logic 1 to the AAC bit in the CONCLR register clears the AA bit. When AA is logic 0, a NACK (HIGH level to SDA) is returned during the acknowledge clock pulse on the SCL line for the following situations:

- A data byte has been received while the I^2C-bus is in the master receiver mode.
- A data byte has been received while the I^2C-bus is in the addressed slave receiver mode.

SI

I^2C-bus interrupt flag. When the I^2C-bus state changes, this bit is set. However, entering state F8 does not set SI since there is nothing for an interrupt handler to do in that case.

While SI is set, the low period of the serial clock on the SCL line is stretched and the serial transfer is suspended. When SCL is HIGH, it is unaffected by the state of the SI lag. Writing a logic 1 to the SIC bit in the CONCLR register resets SI.

STA

START flag. When in master mode, setting this bit causes the I^2C-bus interface to enter master mode and transmit a START, or Repeated-START, condition.

If the bus is free, STA is logic 1, and the I^2C-bus interface is not already in master mode, it enters master mode. It checks the bus and generates a START condition. If the bus is not free, it waits for a STOP condition, which frees the bus. Then, after a delay of a half clock period of the internal clock generator, it generates a START condition. If the I^2C-bus interface is already in master mode and data has been transmitted or received, it transmits a Repeated-START condition. STA may be set at any time, including when the I^2C-bus interface is in an addressed slave mode.

Writing logic 1 to the STAC bit in the CONCLR register clears STA. When STA is logic 0, no START condition or Repeated-START condition is generated.

If STA and STO are both logic 1, a STOP condition is transmitted on the I^2C-bus if the interface is in master mode. It transmits a START condition thereafter. If the I^2C-bus interface is in slave mode, an internal STOP condition is generated but is not transmitted on the bus.

STO

STOP flag. Setting this bit causes the I^2C-bus interface to transmit a STOP condition in master mode or recover from an error condition in slave mode. When STO is logic 1 in master mode, a STOP condition is transmitted on the I^2C-bus. When the bus detects the STOP condition, STO is cleared automatically.

In slave mode, setting this bit can recover from an error condition. In this case, no STOP condition is transmitted to the bus. The hardware behaves as if a STOP condition has been received and it switches to ’not addressed’ slave receiver mode. Hardware automatically clears the STO flag.

I2EN

I^2C-bus interface enable. When I2EN is logic 1, the I^2C-bus interface is enabled. Writing logic 1 to the I2ENC bit in the CONCLR register clears I2EN. When I2EN is logic 0, the I^2C-bus interface is disabled.

When I2EN is logic 0, the SDA and SCL input signals are ignored, the I^2C-bus block is in the not-addressed slave state. The STO bit is forced to logic 0.

Do not use I2EN to release the I²C-bus temporarily since, when I2EN is reset, the I²C-bus status is lost. The AA flag must be used instead.

19.7.2 I²C status register (STAT)

The I²C-bus status register reflects the condition of the corresponding I²C-bus interface. The I²C-bus status register is read-only.

Table 205. STAT register (address 0x4000 0004) bit description

Bit	Symbol	Description	
2:0	-	reserved	0
7:3	Status	actual status information of the I²C-bus interface	11111b
31:8	-	reserved	0

The three least significant bits are always logic 0. Taken as a byte, the status register contents represent a status code. There are 26 possible status codes. When the status code is 0xF8, no relevant information is available and the SI bit is not set. All other 25 status codes correspond to predefined I²C-bus states. When any of these states entered, the SI bit is set. For a complete list of status codes, see Table 221 to Table 226.

19.7.3 I²C-bus data register (DAT)

This register contains the data to be transmitted or the data most recently received. The CPU can only read and write to this register while not in the process of shifting a byte, when the SI bit is set. Data in DAT register remains stable as long as the SI bit is set. Data in DAT register is always shifted from right to left; the first bit to be transmitted is the MSB, bit 7. After a byte has been received, the first bit of received data is in the MSB of the DAT register.

Table 206. DAT register (address 0x4000 0008) bit description

Bit	Symbol	Description	Reset value
7:0	Data	data values that have been received or waiting for transmission	0
31:8	-	reserved	0

19.7.4 I²C-bus slave address register 0 (ADR0)

This register is readable and writable. It is only used when an I²C-bus interface is set to slave mode. In master mode, this register has no effect. The LSB of the ADR register is the general call bit. When this bit is set, the general call address (0x00) is recognized. If this register contains 0x00, the I²C-bus does not acknowledge any address on the bus. All four registers (ADR0 to ADR3) are cleared to this disabled state on reset (see Table 213).

Table 207. ADR0 register (address 0x4000 000C) bit description

Bit	Symbol	Description	Reset value
0	GC	general call enable bit	0
7:1	Address	I²C-bus device address in slave mode	0
31:8	-	reserved	0

19.7.5 I²C-bus SCL high and low duty cycle registers (SCLH / SCLL)

The I2C_PCLK runs at a constant 8 MHz clock rate.

Table 208. SCLH register (address 0x4000 0010) bit description

Bit	Symbol	Description	Reset value
15:0	SCLH	count for SCL high time period selection	0
31:16	-	reserved	0

Table 209. SCLL register (address 0x4000 0014) bit description

Bit	Symbol	Description	Reset value
15:0	SCLH	count for SCL low time period selection	0
31:16	-	reserved	0

To select the appropriate data rate and duty cycle, software must set values for the registers SCLH and SCLL. SCLH defines the number of I2C_PCLK cycles for the SCL HIGH time, SCLL defines the number of I2C_PCLK cycles for the SCL low time. The following formula determines the frequency (I2CPCLK is the frequency of the peripheral I²C-bus clock):

$$I2Cfreq = \frac{I2CPCLK}{SCLH + SCLL} \quad (2)$$

The values for SCLL and SCLH must ensure that the data rate is in the appropriate I²C-bus data rate range. Table 210 gives some examples of I²C-bus rates based on PCLK frequency and SCLL and SCLH values.

Table 210. SCLL + SCLH values for selected I²C-bus clock rates

I²C-bus mode	I²C-bus bit frequency	I2C_PCLK = 8 MHz[1]
Standard mode	100 kHz	80
Fast-mode	400 kHz	20

[1] Restrictions apply based on the Arm clock

SCLL and SCLH values do not necessarily have to be the same. Software can set different duty cycles on SCL by setting these two registers. For example, the I²C-bus specification defines the SCL low time and high time at different values for a fast-mode I²C-bus.

19.7.6 I²C-bus control clear register (CONCLR)

The CONCLR register control clearing of bits in the I²C-bus control register (CON) that controls operation of the I²C-bus interface. Writing a logic 1 to a bit of this register causes the corresponding bit in the ICON register to be cleared. Writing a logic 0 has no effect.

Table 211. CONCLR register (address 0x4000 0018) bit description

Bit	Symbol	Description	Reset value
1:0	-	reserved	0
2	AAC	assert acknowledge flag	0
3	SIC	I²C-bus interrupt clear bit	0

Table 211. CONCLR register (address 0x4000 0018) bit description...*continued*

Bit	Symbol	Description	Reset value
4	-	reserved	0
5	STAC	START flag clear bit	0
6	I2ENC	I^2C-bus interface disable bit	0
31:7	-	reserved	-

AAC

Assert acknowledge clear bit. Writing a logic 1 to this bit clears the AA bit in the CONSET register.

SIC

I^2C-bus interrupt clear bit. Writing a logic 1 to this bit clears the SIC bit in the CONSET register.

STAC

START flag clear bit. Writing a logic 1 to this bit clears the STAC bit in the CONSET register.

I2ENC

I^2C-bus interface disable bit. Writing a logic 1 to this bit clears the I2EN bit in the CONSET register.

19.7.7 I^2C-bus monitor mode control register (MMCTRL)

This monitor mode cannot detect the ACK bit.

Table 212. MMCTRL register (address 0x4000 001C) bit description

Bit	Symbol	Description		Reset value
0	MM_ENA	Monitor mode enable		0
		0	Monitor mode disabled	
		1	The I^2C-bus module enters monitor mode. In this mode, the SDA output is forced HIGH. The I^2C-bus module is prevented from outputting data of any kind (including ACK) onto the I^2C-bus data bus. Depending on the state of the ENA_SCL bit, the output may be also forced high. This action prevents that the module has control over the I^2C-bus clock line.	
1	ENA_SCL	SCL output enable		0
		0	When this bit is cleared to logic 0, the SCL output is forced high when the module is in monitor mode. As described above, this action prevents that the module has any control over the I^2C-bus clock line.	
		1	When this bit is set, the I^2C-bus module may exercise the same control over the clock line that it does in normal operation. So, by acting as a slave peripheral, the I^2C module stretches the clock line (holds it low) until it has had time to respond to an I^2C-bus interrupt.	

Table 212. MMCTRL register (address 0x4000 001C) bit description...*continued*

Bit	Symbol	Description		Reset value
2	MATCH_ALL	select interrupt register match		0
		0	When this bit is cleared, an interrupt is only generated when a match occurs to one of the (up-to) 4 address registers described above. That is, the module responds as a normal slave as far as address-recognition is concerned.	
		1	When this bit is set to logic 1 and the I^2C-bus is in monitor mode, an interrupt is generated on any address received. This interrupt generation enables the part to monitor all traffic on the bus.	
31:3	-	reserved		0

Note: *If the MM_ENA is logic 0 (that is, if the module is not in monitor mode), the ENA_SCL and MATCH_ALL bits have no effect.*

Interrupt in monitor mode

When the module is in monitor mode, all interrupts occur as normal. The first interrupt occurs when an address match is detected. An address match is address received if the MATCH_ALL bit is set or an address matching one of the four address registers.

Subsequent to an address match detection, interrupts are generated in the following cases: After each data byte is received for a slave-write transfer or after each byte that the module 'thinks' it has transmitted for a slave-read transfer. In this second case, the data register actually contains data transmitted by another slave on the bus addressed by the master.

Following all of these interrupts, the processor may read the data register to see what was transmitted on the bus.

Loss of arbitration in monitor mode

In monitor mode, the I^2C-bus module is unable to respond to a request for information by the bus master or issue an ACK. Some other slave on the bus responds instead. The result is a lost-arbitration state as far as our module is concerned.

Software must be aware of the fact that the module is in monitor mode and must not respond to any loss of arbitration state detected. In addition, hardware may be designed into the module to block some/all loss of arbitration states from occurring. If those states would either prevent a desired interrupt from occurring or cause an unwanted interrupt to occur, this design is necessary.

19.7.8 I^2C-bus slave address registers 1:3 (ADR1/2/3)

This register is readable and writable. It is only used when an I^2C-bus interface is set to slave mode. In master mode, this register has no effect. The LSB of the ADR register is the general call bit. When this bit is set, the general call address (0x00) is recognized. If this register contains 0x00, the I^2C-bus does not acknowledge any address on the bus. All four registers (ADR0 to ADR3) are cleared to this disabled state on reset (see Table 207).

Table 213. ADR1/2/3 registers (addresses 0x4000 0020,0024,0028) bit description

Bit	Symbol	Description	Reset value
0	GC	General call enable bit	0
7:1	address	I^2C-bus device address in slave mode	0
31:8	-	reserved	0

19.7.9 I^2C-bus data buffer register (DATA_BUFFER)

In monitor mode, the I^2C-bus module may lose the ability to stretch the clock (stall the bus) if the ENA_SCL bit is not set. In this case, the processor has a limited amount of time to read the contents of the data received on the bus. If the processor reads the DAT shift register as normal, it may have only one bit-time to respond to the interrupt before the received data is overwritten by new data.

To give the processor more time to respond, a new 8-bit, read-only DATA_BUFFER register is added. The contents of the 8 MSBs of the DAT shift register are transferred to the DATA_BUFFER automatically after every 9 bits have been received. The 9 bits being 8 bits of data plus ACK or NACK. This action means that, to the interrupt and read the data before it is overwritten, the processor has 9-bit transmission times to respond.

The processor is still able to read the DAT register directly, as usual, and the behavior of DAT is not altered in any way.

Although the DATA_BUFFER register is primarily intended for use in monitor mode with the ENA_SCL bit = logic 0, it is available for reading at any time under any mode of operation.

Table 214. DATA_BUFFER register (address 0x4000 002C) bit description

Bit	Symbol	Description	Reset value
7:0	data	contents of the 8 MSBs of the DAT shift register	0
31:8	-	reserved	0

19.7.10 I^2C-bus mask registers 0:3 (MASK0/1/2/3)

The four mask registers each contain seven active bits (7:1). Active bits set to logic 1, cause an automatic compare. It compares the corresponding bit of the received address with the ADDRn register associated with that mask register. In other words, when determining an address match, bits in an ADDRn register which are masked are disregarded.

On reset, all mask register bits are cleared to logic 0.

The mask register has no effect on comparison to the general call address (‘000 0000’).

Bits(31:8) and bit(0) of the mask registers are unused and must not be written to. These bits are always read back as logic 0s.

Table 215. MASK0/1/2/3 register (addresses 0x4000 0030,0034,0038,003C) bit description

Bit	Symbol	Description	Reset value
0	-	reserved	0
7:1	data	I^2C-bus device address in slave mode	0
31:8	-	reserved	-

19.8 I²C-bus operating modes

In a given application, the I²C-bus block may operate as a master, a slave, or both. In the slave mode, the I²C-bus hardware looks for any one of its four slave addresses and the general call address. If one of these addresses is detected, an interrupt is requested. If the processor wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered. This wait state is to ensure that a possible slave operation is not interrupted. If bus arbitration is lost in master mode, the I²C-bus block immediately switches to slave mode. It can then detect its own slave address in the same serial transfer.

19.8.1 Master transmitter mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 32). Before the master transmitter mode can be entered. the CONSET register must be initialized as follows:

Table 216. CONSET used to initialize master transmitter mode

BIT	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	X	-	-

Also, the I²C-bus rate must be configured in the SCLL and SCLH registers. To enable the I²C-bus block, I2EN must be set to logic 1. If the AA bit is reset, the I²C-bus block does not acknowledge its own slave address. If another device becomes master of the bus, it does not acknowledge the general call address. In other words, if AA is reset, the I²C-bus interface cannot enter slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit. The I²C logic tests the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set and the status code in the status register (STAT) is 0x08. The interrupt handler uses this status code to enter the appropriate state service routine. This routine loads DAT with the slave address and the data direction bit (SLA+W). The SI bit in CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again. A number of status codes in STAT is possible. 0x18, 0x20, or 0x38 are for the master mode and also 0x68, 0x78, or 0xB0 if the slave mode was enabled (AA = logic 1).Table 217 details the appropriate action to be taken for each of these status codes. After a Repeated-START condition (state 0x10), the I²C-bus block may switch to the master receiver mode by loading DAT with SLA+R).

Table 217. Master transmitter mode

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C hardware
0x08	a START condition has been transmitted	load SLA+W; Clear STA	X	0	0	X	SLA+W is transmitted; ACK bit is received.
0x10	a Repeated-START condition has been transmitted	load SLA+W	X	0	0	X	as above
		load SLA+R; Clear STA	X	0	0	X	SLA+W is transmitted; the I^2C-bus block is switched to master receiver mode.
0x18	SLA+W has been transmitted; ACK has been received	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x20	SLA+W has been transmitted; NACK has been received	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x28	data byte in I2DAT has been transmitted; ACK has been received.	load data byte	0	0	0	X	Data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x30	data byte in I2DAT has been transmitted; NOT ACK has been received.	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received
		no I2DAT action	1	0	0	X	repeated START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.

Table 217. Master transmitter mode *...continued*

Status code I2CSTAT	Status of the I²C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I²C hardware
0x38	arbitration lost in SLA +R/W or Data bytes	no I2DAT action	0	0	0	X	I²C-bus is released; not addressed, slave mode is entered.
		no I2DAT action	1	0	0	X	a START condition is transmitted when the bus becomes free.

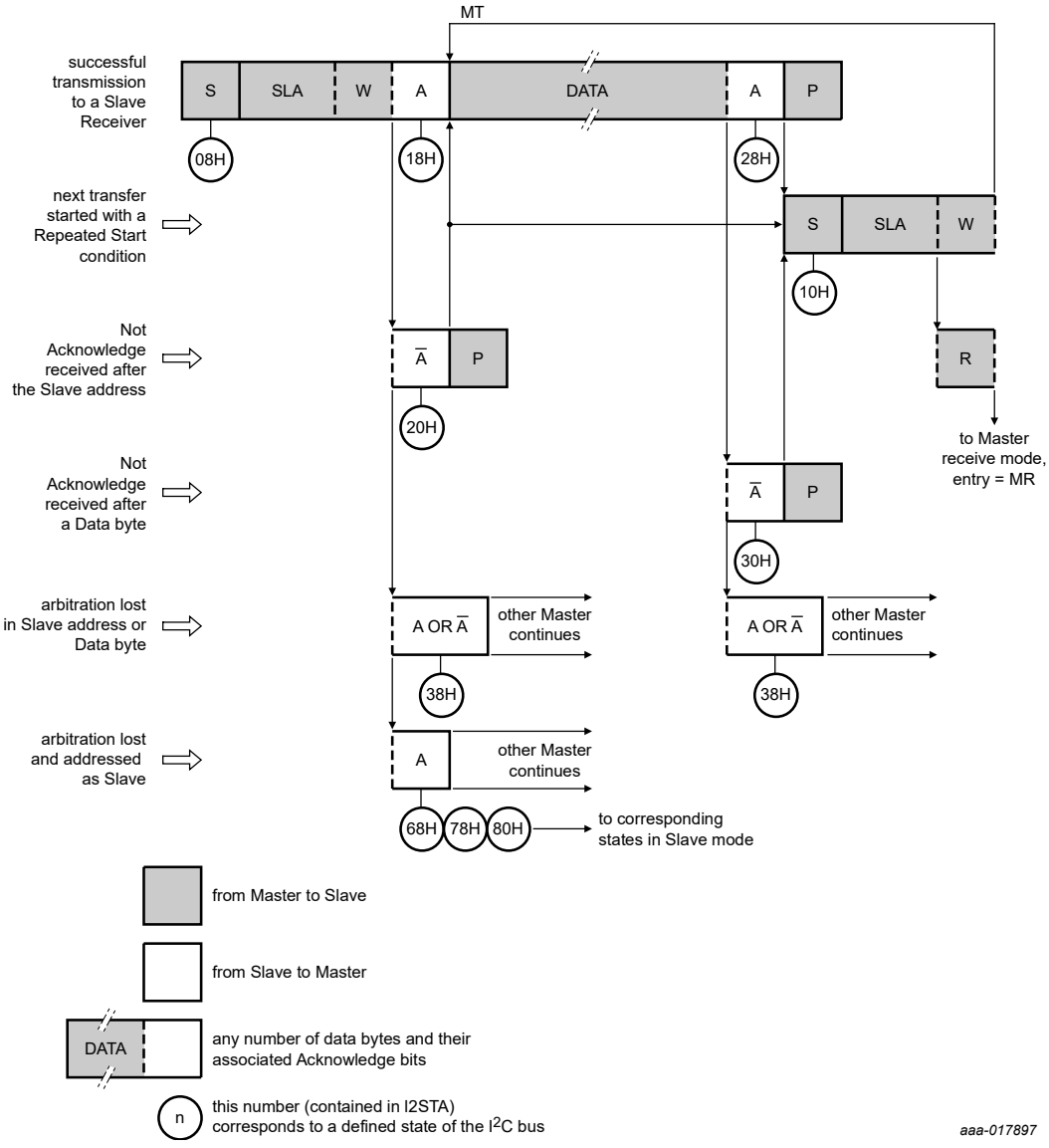


Figure 32. Format and states in the master transmitter mode

19.8.2 Master receiver mode

In the master receiver mode, data is received from a slave transmitter. The transfer is initiated in the same way as in the master transmitter mode. When the START condition has been transmitted, the interrupt handler must load the slave address and data direction bit to the I^2C-bus Data register. It must then clear the SI bit. In this case, the data direction bit (R/W) must be logic 1 to indicate a read.

When the slave address and data direction bit have been transmitted, and acknowledged, the SI bit is set and the status register shows the status code. For master mode, the possible status codes are 0x40, 0x48, or 0x38. For slave mode, the possible status codes are 0x68, 0x78, or 0xB0. For details, see Table 222.

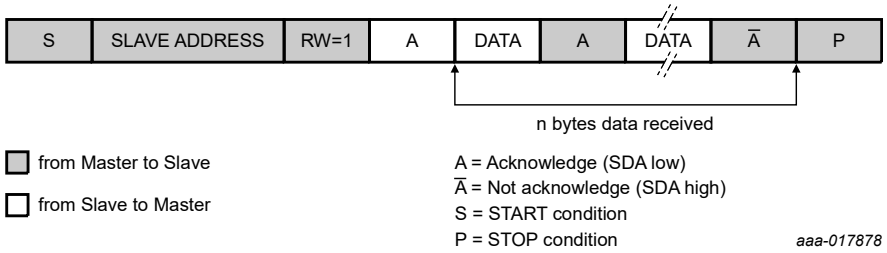


Figure 33. Data format of master receiver mode

After a Repeated-START condition, I^2C-bus may switch to the master transmitter mode:

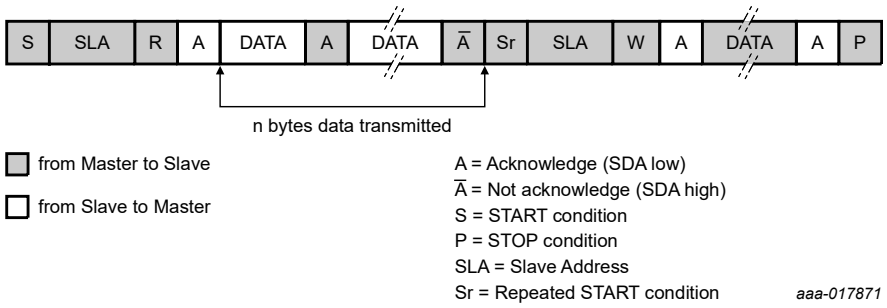


Figure 34. Master receiver switches to master transmitter after sending Repeated-START

19.8.3 Slave receiver mode

In the slave receiver mode, data bytes are received from a master transmitter. To initialize the slave receiver mode, write any of the slave address registers (ADR0-3) and write the I^2C-bus control set register (CONSET) as shown in Table 218.

Table 218. CONSET used to configure slave mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

To enable the I^2C-bus function, I2EN must be set to logic 1. To acknowledge its own slave address or the general call address, the AA bit must be set to logic 1. The STA, STO, and SI bits are set to logic 1.

After ADR and CONSET are initialized, the I^2C-bus interface waits until addressed by its own address or general address followed by the data direction bit. If the direction bit is logic 0 (write), it enters slave receiver mode. If the direction bit is logic 1 (read), it enters slave transmitter mode. After the address and direction bit have been received, the SI bit is set. A valid status code can now be read from the status register (STAT). Refer to for the status codes and actions.

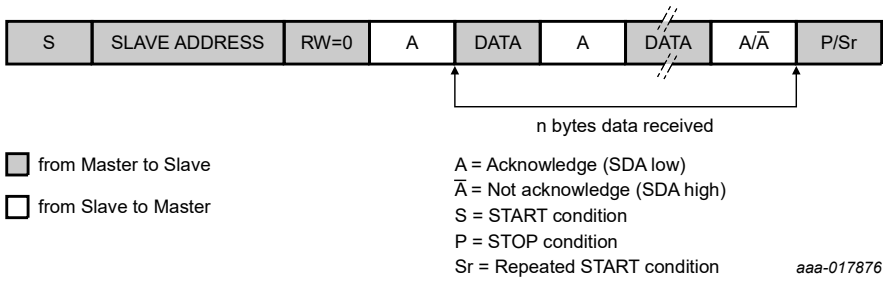


Figure 35. Data format of slave receiver mode

19.8.4 Slave transmitter mode

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit is logic 1, indicating a read operation. Serial data is transmitted via SDA, while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. In a given application, the I^2C-bus may operate as a master and as a slave.

In the slave mode, the I^2C-bus hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered. This wait state is to ensure that a possible slave action is not interrupted. If bus arbitration is lost in master mode, the I^2C-bus block immediately switches to slave mode. It can then detect its own slave address in the same serial transfer.

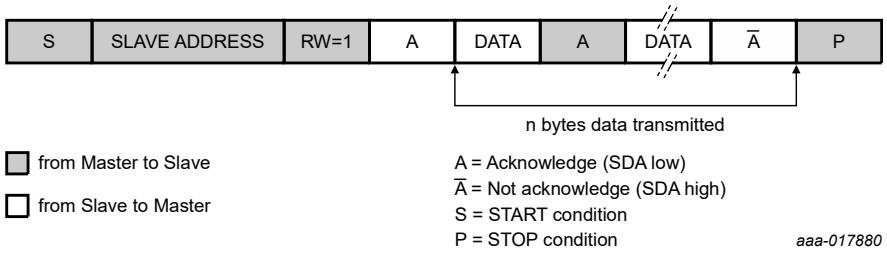


Figure 36. Data format of slave mode

19.9 I²C-bus implementation and operation

Figure 37 shows how the on-chip I²C-bus interface is implemented. The following text describes the individual blocks.

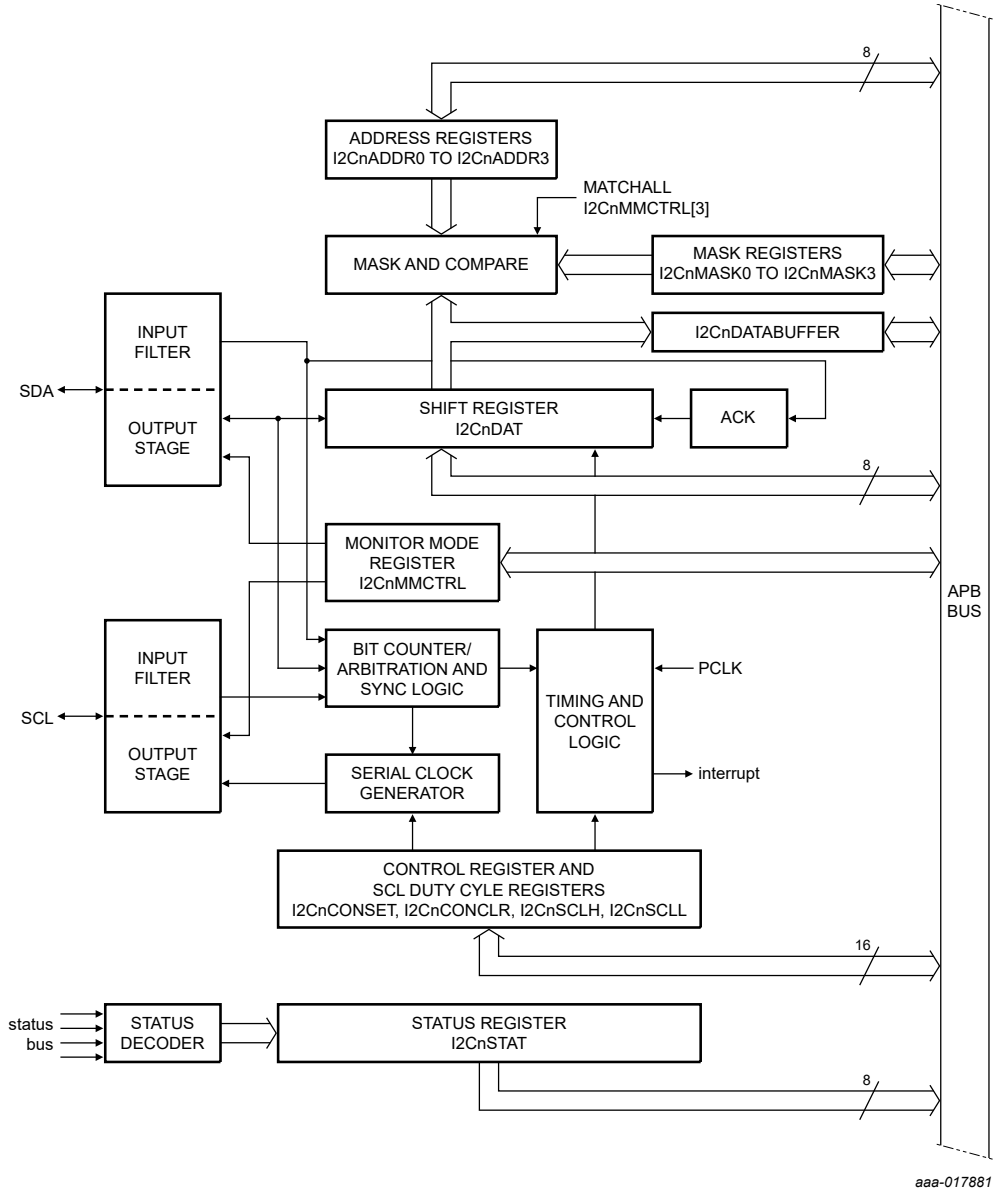


Figure 37. I²C serial interface block diagram

19.9.1 Input filters and output stages

Input signals are synchronized with the internal clock. Spikes shorter than three clock cycles are filtered out.

The output for I^2C-bus is a special pad designed to conform to the I^2C-bus specification.

19.9.2 Address registers, ADDR0 to ADDR3

These registers may be loaded with the 7-bit slave address (7 MSBs) to which the I^2C-bus block responds when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable general call address (0x00) recognition. When multiple slave addresses are enabled, the actual address received may be read from the DAT register at the state where the slave address has been received.

19.9.3 Address mask registers, MASK0 to MASK3

The four mask registers each contain seven active bits (7:1). Any bit in these registers which is set to logic 1, causes an automatic compare. It compares the corresponding bit of the received address with the ADDRn register associated with that mask register. In other words, when determining an address match, bits in an ADDRn register which are masked are not considered.

When an address match interrupt occurs, the processor has to read the data register (DAT) to determine what the received address was that caused the match.

19.9.4 Comparator

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in ADR). It also compares the first received 8-bit byte with the general call address (0x00). If an equality is found, the appropriate status bits are set and an interrupt is requested.

19.9.5 Shift register, DAT

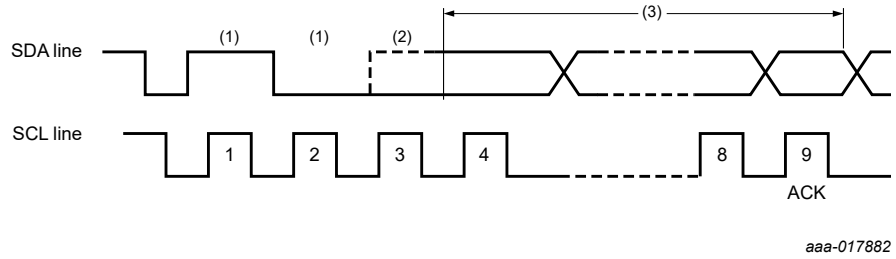
This 8-bit register contains a byte of serial data to be transmitted or a byte which has been received. Data in DAT is always shifted from right to left. The first bit transmitted is the MSB (bit 7). After a byte has been received, the first bit of received data is at the MSB of DAT. While data is shifted out, data on the bus is simultaneously being shifted in. DAT always contains the last byte present on the bus. So, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in DAT.

19.9.6 Arbitration and synchronization logic

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I^2C-bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost. The I^2C-bus block immediately changes from master transmitter to slave receiver. The I^2C-bus block continues to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while the I^2C-bus block is returning a ’not acknowledge’ (logic 1) to the bus. When another device on the bus pulls this signal low, arbitration is lost. Since it can

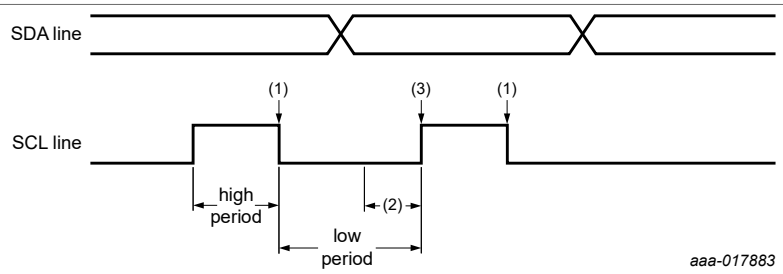
occur only at the end of a serial byte, the I²C-bus block generates no further clock pulses. Figure 38 shows the arbitration procedure.



1. Another device transmits serial data.
2. Another device overrules a logic (dotted line) transmitted this I²C-bus master by pulling the SDA line low. Arbitration is lost and this I²C-bus enters slave receiver mode.
3. This I²C-bus is in slave receiver mode but still generates clock pulses until the current byte has been transmitted. This I²C-bus does not generate clock pulses for the next byte. Data on SDA originates from the new master once it has won arbitration.

Figure 38. Arbitration procedure

The synchronization logic synchronizes the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the following occurs. The device that generates the shortest 'marks' determines the 'mark' duration. The device that generates the longest 'spaces' determines the 'space' duration. Figure 39 shows the synchronization procedure.



1. Another device pulls the SCL line low before this I²C-bus has timed a complete high time. The other device effectively determines the (shorter) HIGH period.
2. Another device continues to pull the SCL line low after this I²C-bus has timed a complete low time and released SCL. The I²C-bus clock generator is forced to wait until SCL goes HIGH. The other device effectively determines the (longer) low period.
3. The SCL line is released and the clock generator begins timing the HIGH time.

Figure 39. Serial clock synchronization

To slow down the bus master, a slave may stretch the space duration. The space duration may also be stretched for handshaking purposes. This stretching can be done after each bit or after a complete byte transfer. The I²C-bus block stretches the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set and the stretching continues until the serial interrupt flag is cleared.

19.9.7 Serial clock generator

When the I²C-bus block is in the master transmitter or master receiver mode, this programmable clock pulse generator provides the SCL clock pulses. When the I²C-bus

block is in slave mode, it is switched off. The I^2C-bus output clock frequency and duty cycle are programmable via the I^2C-bus clock control registers. See the description of the I2CSCLL and I2CSCLH registers for details. The output clock pulses have a duty cycle as programmed unless the bus is synchronizing with other SCL clock sources as described above.

19.9.8 Timing and control

The timing and control logic generates the timing and control signals for serial byte handling. This logic block performs the following tasks:

- Provides the shift pulses for DAT
- Enables the comparator
- Generates and detects START and STOP conditions
- Receives and transmits acknowledge bits
- Controls the master and slave modes
- Contains interrupt request logic
- Monitors the I^2C-bus status

19.9.9 Control register; CONSET and CONCLR

The I^2C-bus control register contains bits used to control the following I^2C block functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

The contents of the I^2C-bus control register may be read as CONSET. Writing to CONSET sets bits in the I^2C-bus control register that correspond to ones in the value written. Conversely, writing to CONCLR clears bits in the I^2C-bus control register that correspond to ones in the value written.

19.9.10 Status decoder and status register

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I^2C-bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. If all four modes of the I^2C-bus block are used, there are 26 possible bus states. When the serial interrupt flag is set (by hardware), the 5-bit status code is latched into the five most significant bits of the status register. It remains stable until software clears the interrupt flag. The three least significant bits of the status register are always logic 0. If the status code is used as a vector to service routines, then eight address locations displace the routines. 8 bytes of code is sufficient for most of the service routines.

19.10 Details of I²C-bus operating modes

As mentioned above, there are four operating modes. These modes are:

- Master transmitter
- Master receiver
- Slave receiver
- Slave transmitter

Figure 41, Figure 42, Figure 43, and Section 19.10.6.1 show data transfers in each mode of operation.

Table 219 lists abbreviations used in these figures when describing the I²C-bus operating modes.

Table 219. Abbreviations used to describe an I²C-bus operation

Abbreviation	Explanation
S	START condition
SLA	7-bit slave address
R	read bit (HIGH level at SDA)
W	write bit (low level at SDA)
A	acknowledge bit (low level at SDA)
$\overline{A}$	not acknowledge bit (high level at SDA)
Data	8-bit data byte
P	STOP condition

In Table 221 to Figure 44, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the STAT register. At these points, to continue or complete the serial transfer, a service routine must be executed. These service routines are not critical since the serial transfer is suspended until software clears the serial interrupt flag.

When a serial interrupt routine is entered, the status code in STAT is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in tables from Table 232 to Table 238.

19.10.1 Master transmitter mode

In the master transmitter mode, a number of data bytes is transmitted to a slave receiver (see Figure 40). Before the master transmitter mode can be entered. the CONSET register must be initialized as follows:

Table 220. CONSET used to initialize master transmitter mode

BIT	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	X	-	-

Also, the I²C-bus rate must be configured in the SCLL and SCLH registers. To enable the I²C-bus block, I2EN must be set to logic 1. If the AA bit is reset, the I²C-bus block does not acknowledge its own slave address. Ifof another device becomes master of the bus, it

does not acknowledge the general call address. In other words, if AA is reset, the I^{2}C-bus interface cannot enter slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit. The I^{2}C logic tests the I^{2}C-bus and generate a START condition when the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (STAT) is 0x08. This status code is used by the interrupt handler to enter the appropriate state service routine. This routine loads DAT with the slave address and the data direction bit (SLA+W). The SI bit in CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again. A number of status codes in STAT is possible. There are 0x18, 0x20, or 0x38 for the master mode and 0x68, 0x78, or 0xB0 also if the slave mode was enabled (AA = logic 1). Table 221 details the appropriate action to be taken for each of these status codes. After a Repeated-START condition (state 0x10). The I^{2}C-bus block may switch to the master receiver mode by loading DAT with SLA+R).

Table 221. Master transmitter mode

Status code I2CSTAT	Status of the I^{2}C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^{2}C hardware
0x08	a START condition has been transmitted	load SLA+W; Clear STA	X	0	0	X	SLA+W is transmitted; ACK bit is received.
0x10	a Repeated-START condition has been transmitted	load SLA+W	X	0	0	X	as above
		load SLA+R; Clear STA	X	0	0	X	SLA+W is transmitted; the I^{2}C-bus block is switched to Master receiver mode.
0x18	SLA+W has been transmitted; ACK has been received	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x20	SLA+W has been transmitted; NACK has been received	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.

Table 221. Master transmitter mode *...continued*

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C hardware
0x28	data byte in I2DAT has been transmitted; ACK has been received.	load data byte	0	0	0	X	Data byte is transmitted; ACK bit is received.
		no I2DAT action	1	0	0	X	Repeated-START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x30	data byte in I2DAT has been transmitted; NOT ACK has been received.	load data byte	0	0	0	X	data byte is transmitted; ACK bit is received
		no I2DAT action	1	0	0	X	repeated START is transmitted.
		no I2DAT action	0	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset.
0x38	arbitration lost in SLA +R/W or Data bytes	no I2DAT action	0	0	0	X	I^2C-bus is released; not addressed, slave mode is entered.
		no I2DAT action	1	0	0	X	a START condition is transmitted when the bus becomes free.

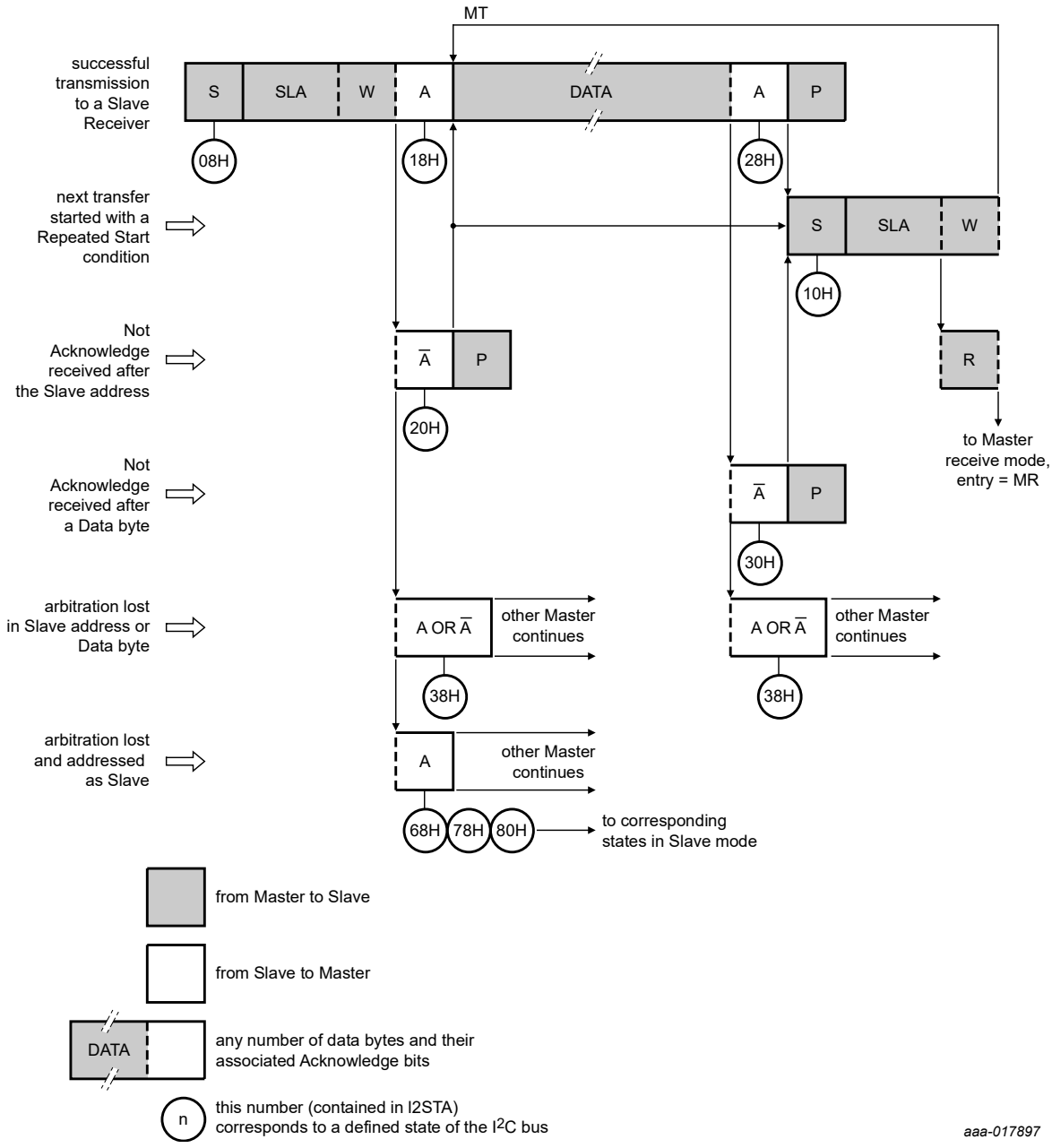


Figure 40. Format and states in the master transmitter mode

19.10.2 Master receiver mode

In the master receiver mode, a number of data bytes is received from a slave transmitter (see Figure 41). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt handler must load DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in CON must then be cleared before the serial transfer can continue.

When the slave address and data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again. A number of status codes in STAT is then possible. If the slave mode was enabled

(AA = 1), these codes are 0x40, 0x48, or 0x38 and also 0x68, 0x78, or 0xB0 for the master mode. Table 222 details the appropriate action to be taken for each of these status codes. After a Repeated-START condition (state 0x10), the I^2C-bus block may switch to the master transmitter mode by loading DAT with SLA+W.

Table 222. Master receive mode

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C hardware
0x08	A START condition has been transmitted	load SLA+W	X	0	0	X	SLA+R is transmitted; ACK bit is received
0x10	A Repeated-START condition has been transmitted	load SLA+R	X	0	0	X	as above
		load SLA+W	X	0	0	X	SLA+W is transmitted; the I^2C block is switched to Master receiver mode
0x38	Arbitration lost in NACK bit	no I2DAT action	0	0	0	X	I^2C-bus is released; not addressed slave mode is entered
		no I2DAT action	1	0	0	X	a START condition is transmitted when the bus becomes free
0x40	SLA+R has been transmitted; ACK has been received	load data byte	0	0	0	0	data byte is received; NACK bit is returned
		no I2DAT action	0	0	0	1	data byte is received; ACK bit is returned
0x48	SLA+R has been transmitted; NACK has been received	no I2DAT action	1	0	0	X	repeated START is transmitted
		no I2DAT action	0	1	0	X	STOP condition is transmitted; STO flag is reset
		no I2DAT action	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset
0x50	Data byte has been received; ACK has been returned.	read data byte	0	0	0	0	data byte is received; NACK bit is returned
		read data byte	0	0	0	1	data byte is received; ACK bit is returned
0x58	Data byte has been received; NACK has been returned.	read data byte	1	0	0	X	repeated START condition is transmitted
		read data byte	0	1	0	X	a STOP condition is transmitted; STO flag is reset
		read data byte	1	1	0	X	STOP condition followed by a START condition is transmitted; STO flag is reset

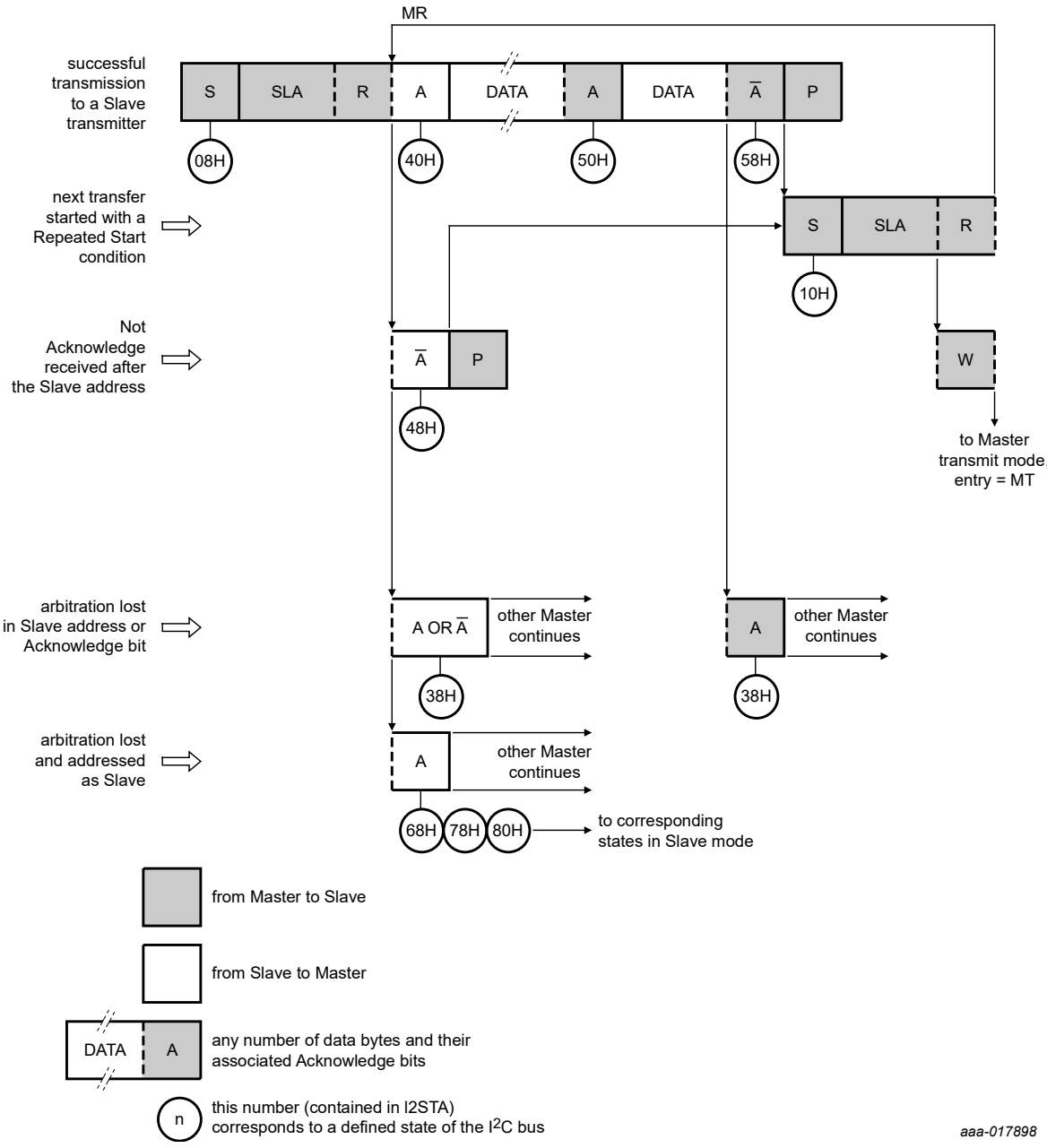


Figure 41. Format and states in Master receiver mode

19.10.3 Slave receiver mode

In the slave receiver mode, a number of data bytes is received from a master transmitter (see Figure 42). To initiate the slave receiver mode, ADR and CON must be loaded as follows:

Table 223. ADR usage in slave receiver mode

BIT	7 - 1	0
Symbol	own slave 7-bit address	GC

The upper 7 bits are the address to which the I2C-bus block responds when addressed by a master. If the LSB (GC) is set, the I2C-bus block responds to the general call address (0x00). Otherwise, it ignores the general call address.

Table 224. CONSET used to initialize slave transmitter mode

BIT	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

The I2C-bus rate settings do not affect the I2C-bus block in the slave mode. To enable the I2C block, I2EN must be set to logic 1. To enable the I2C-bus block to acknowledge its own slave address or the general call address, the AA bit must be set. STA, STO, and SI must be reset.

When ADR and CON have been initialized, the I2C block waits until addressed by its own slave address followed by the data direction bit. This bit must be logic 0 (write) for the I2C block to operate in the slave receiver mode. After the I2C-bus block has received its own slave address and the W bit, the serial interrupt flag (SI) is set. A valid status code can be read from STAT. This status code is used to vector to a state service routine. Table 225 details the appropriate action to be taken for each of these status codes. If arbitration is lost while the I2C-bus block is in the master mode, the slave receiver mode may also be entered (see status 0x68 and 0x78).

Is the AA bit is reset during a transfer, the I2C block returns a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, the I2C-bus block does not respond to its own slave address or a general call address. However, the I2C-bus is still monitored and address recognition may be resumed at any time by setting AA. So, the AA bit may be used to isolate the I2C-bus block from the I2C-bus temporarily.

Table 225. Slave receiver mode

Status code I2CSTAT	Status of the I2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I2C hardware
0x60	own SLA+W has been received; ACK has been returned.	no I2DAT action	X	0	0	0	data byte is received; NACK is returned.
		no I2DAT action	X	0	0	1	data byte is received; ACK is returned.
0x68	arbitration lost in SLA +R/W as master; Own SLA+W has been received, ACK returned.	no I2DAT action	X	0	0	0	data byte is received; NACK is returned.
		no I2DAT action	X	0	0	1	data byte is received; ACK is returned.
0x70	general call address (0x00) has been received; ACK has been returned.	no I2DAT action	X	0	0	0	data byte is received; NACK is returned.
		no I2DAT action	X	0	0	1	data byte is received; ACK is returned.
0x78	arbitration lost in SLA +R/W as master; general call address has been received,	no I2DAT action	X	0	0	0	data byte is received; NACK is returned.
		no I2DAT action	X	0	0	1	data byte is received; ACK is returned.

Table 225. Slave receiver mode*...continued*

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C hardware
0x80	previously addressed with own slave address; DATA has been received; ACK has been returned.	read data byte	X	0	0	0	data byte is received; NACK is returned.
		read data byte	X	0	0	1	data byte is received; ACK is returned.
0x88	previously addressed with own SLA; DATA byte has been received; NACK has been returned.	read data byte	0	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address.
		read data byte	0	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
		read data byte	1	0	0	0	switched to not addressed slave mode; No recognition of own SLA or general call address. When the bus becomes free, A START condition is transmitted.
		read data byte	1	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
0x90	previously addressed with general call; DATA byte has been received; ACK has been returned.	read data byte	X	0	0	0	data byte is received; NACK is returned.
		read data byte	X	0	0	1	data byte is received; ACK is returned.
0x98	previously addressed with general call; DATA byte has been received; NACK has been returned.	read data byte	0	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address.
		read data byte	0	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
		read data byte	1	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address. When the bus becomes free, a START condition is transmitted.
		read data byte	1	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized. When the bus becomes free, a START condition is transmitted.

Table 225. Slave receiver mode*...continued*

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C hardware
0xA0	a STOP condition or Repeated-START condition has been received while still addressed as slave receiver or slave transmitter.	no STDAT action	0	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address.
		no STDAT action	0	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
		no STDAT action	1	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address. When the bus becomes free, a START condition is transmitted.
		no STDAT action	1	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized. When the bus becomes free, a START condition is transmitted.

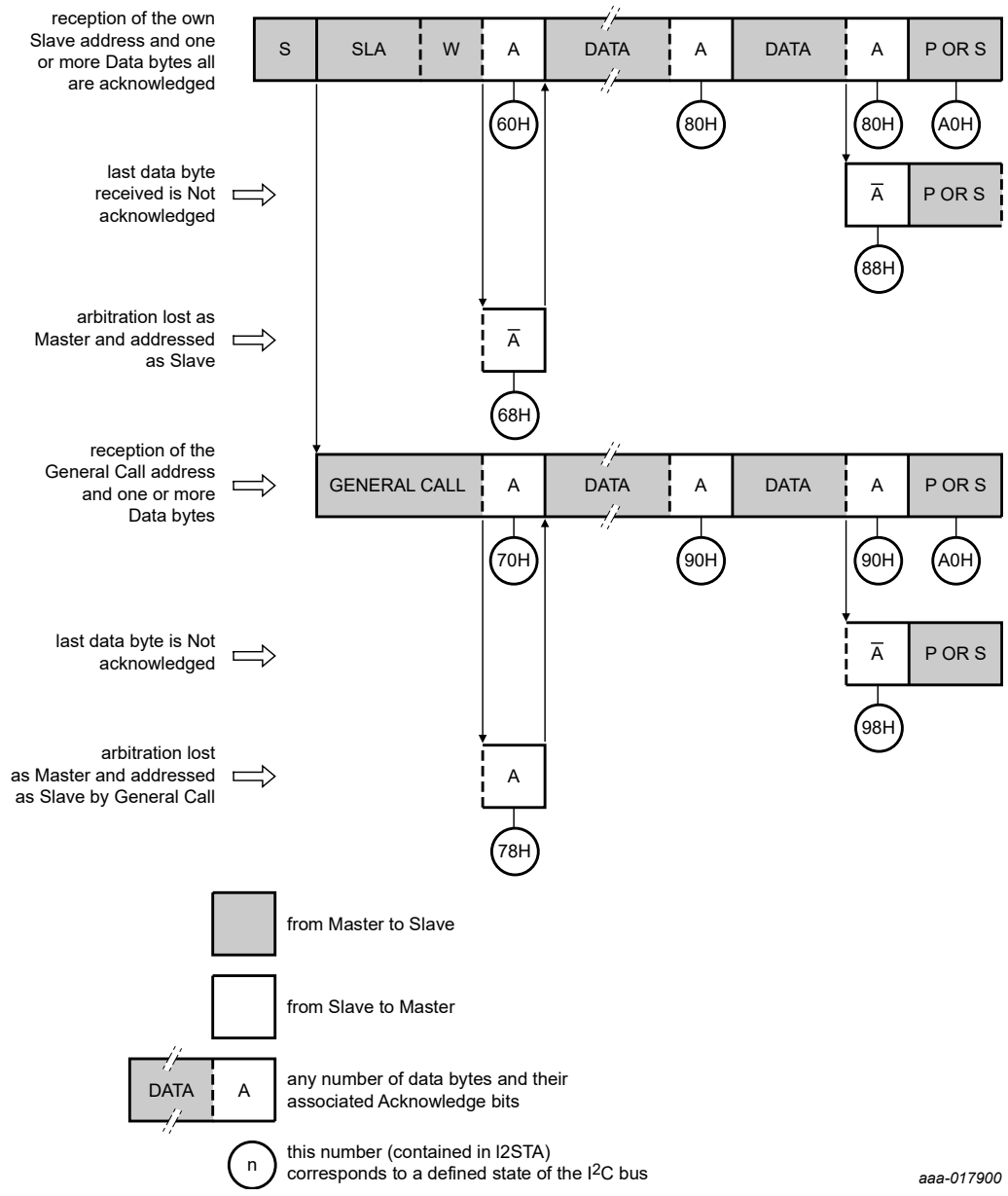


Figure 42. Format and states in slave receiver mode

19.10.4 Slave transmitter mode

In the slave transmitter mode, a number of data bytes is transmitted to a master receiver (see Figure 43). Data transfer is initialized as in the slave receiver mode. When ADR and CON have been initialized, the I^2C-bus block waits until it is addressed by its own slave address followed by the data direction bit. The data direction bit must be logic 1 (read) for the I^2C-bus block to operate in the slave transmitter mode. After the I^2C-bus block has received its own slave address and the R bit, the serial interrupt flag (SI) is set and a valid status code can be read from STAT. This status code is used to vector to a state service routine. Table 226 details the appropriate action to be taken for each of these status codes. If arbitration is lost while the I^2C-bus block is in the master mode (see state 0xB0), the slave transmitter mode may also be entered.

If the AA bit is reset during a transfer, the I^2C-bus block transmits the last byte of the transfer and enters state 0xC0 or 0xC8. If the master receiver continues the transfer, the I^2C block is switched to the not-addressed slave mode and ignores the master receiver.

Thus the master receiver receives all logic 1s as serial data. While AA is reset, the I^2C-bus block does not respond to its own slave address or a general call address. However, the I^2C-bus is still monitored and address recognition may be resumed at any time by setting AA. This action means that the AA bit may be used to isolate the I^2C-bus block from the I^2C-bus temporarily.

Table 226. Slave transmitter mode

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C-bus hardware
0xA8	own SLA+R has been received; ACK has been returned.	load data byte	X	0	0	0	data byte is transmitted; NACK is received.
		load data byte	X	0	0	1	data byte is transmitted; ACK is received.
0xB0	arbitration lost in SLA +R/W as master; Own SLA+W has been received, ACK returned.	load data byte	X	0	0	0	data byte is transmitted; NACK is received.
		load data byte	X	0	0	1	data byte is transmitted; ACK is received.
0xB8	data byte in I2DAT has been transmitted; ACK has been received.	load data byte	X	0	0	0	data byte is transmitted; NACK is received.
		load data byte	X	0	0	1	data byte is transmitted; ACK is received.
0xC0	data byte in I2DAT has been transmitted; NACK has been received.	no I2DAT action	0	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address.
		no I2DAT action	0	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
		no I2DAT action	1	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address. When the bus becomes free, a START condition is transmitted.
		no I2DAT action	1	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized. When the bus becomes free, a START condition is transmitted.

Table 226. Slave transmitter mode *...continued*

Status code I2CSTAT	Status of the I^2C-bus and hardware	Application software response					
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	Next action taken by I^2C-bus hardware
0xC8	last data byte in I2DAT has been transmitted (AA = 0); ACK has been received.	no I2DAT action	0	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address.
		no I2DAT action	0	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized.
		no I2DAT action	1	0	0	0	switched to not addressed slave mode; no recognition of own SLA or general call address. When the bus becomes free, a START condition is transmitted.
		no I2DAT action	1	0	0	1	switched to not addressed slave mode; Own SLA is recognized; If I2ADR[0] = logic 1, general call address is recognized. When the bus becomes free, a START condition is transmitted.

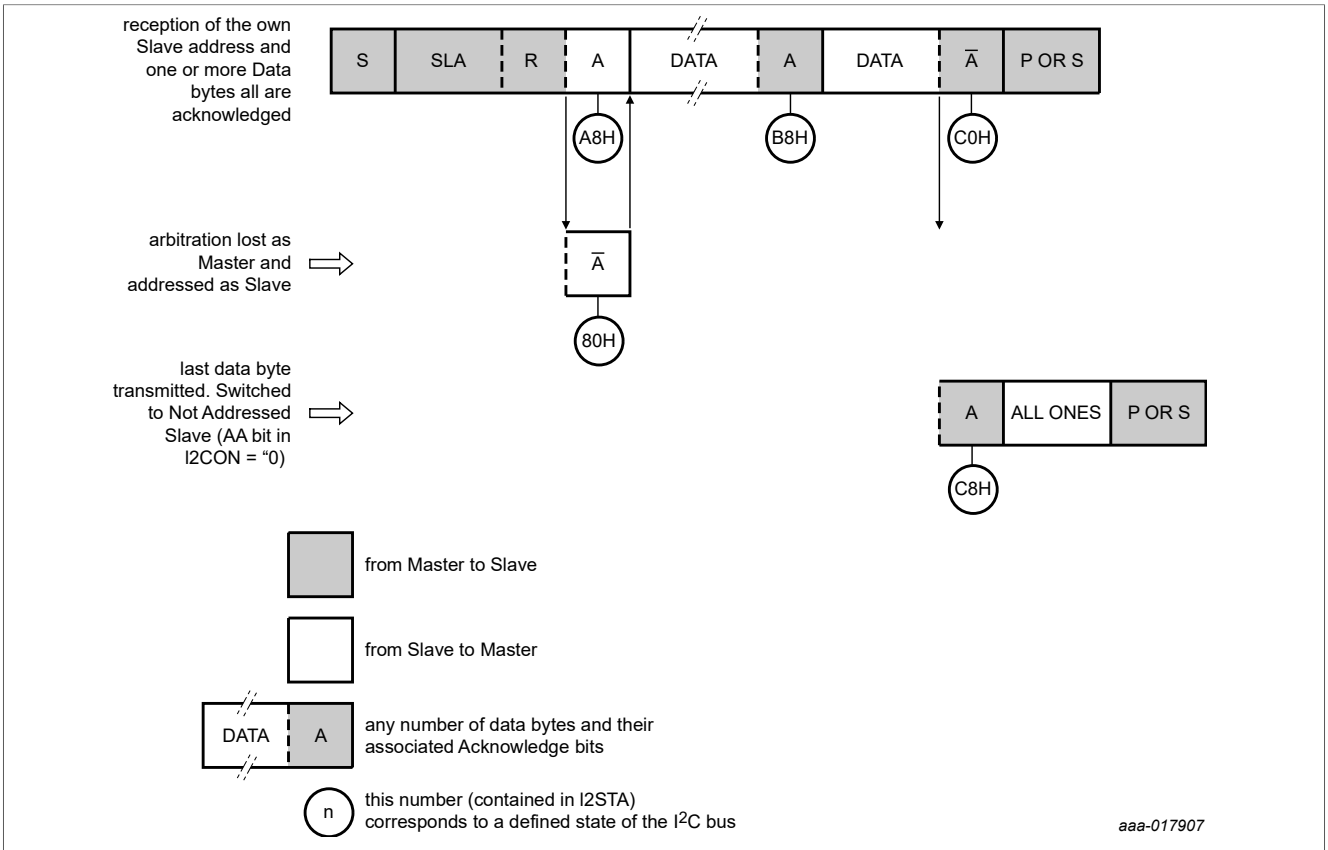


Figure 43. Format and states in slave transmitter mode

19.10.5 Miscellaneous states

There are two STAT codes that do not correspond to a defined I²C-bus hardware state (see Table 227). These codes are discussed below.

Table 227. Miscellaneous states

Status code (STAT)	Status of the I²C-bus and hardware	Application software response					
		To/From DAT	To CON				
			STA	STO	SI	AA	Next action taken by I2C-bus hardware
0xF8	no relevant state information available; SI = 0.	no DAT action	no CON action				wait or proceed current transfer.
0x00	bus error during master or selected slave modes, due to an illegal START or STOP condition. State 0x00 can also occur when interference causes the I²C-bus block to enter an undefined state.	no DAT action	0	1	0	X	only the internal hardware is affected in the master or addressed slave mode. In all cases, the bus is released and the I²C-bus block is switched to the not addressed slave mode. STO is reset.

19.10.5.1 STAT = 0xF8

This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This code occurs between other states and when the I²C-bus block is not involved in a serial transfer.

19.10.5.2 STAT = 0x00

This status code indicates that a bus error has occurred during an I²C-bus serial transfer. When a START or STOP condition occurs at an illegal position in the format frame, a bus error is caused. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal I²C-bus block signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This procedure causes the I²C-bus block to enter the 'not addressed' slave mode (a defined state) and to clear the STO flag. No other bits in CON are affected. The SDA and SCL lines are released (a STOP condition is not transmitted).

19.10.6 Some special cases

The I²C-bus hardware has facilities to handle the following special cases that may occur during a serial transfer:

- Simultaneous Repeated-START conditions from two masters
- Data transfer after loss of arbitration
- Forced accesses to the I²C-bus
- I²C-bus obstructed by a low level on SCL or SDA
- Bus error

19.10.6.1 Simultaneous Repeated-START conditions from two masters

A Repeated-START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a

Repeated-START condition (see Figure 44). Until this condition occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the I^2C-bus hardware detects a Repeated-START condition on the I^2C-bus before generating a Repeated-START condition itself, it releases the bus and no interrupt request is generated. If another master frees the bus by generating a STOP condition, the I^2C-bus block transmits a normal START condition (state 0x08). A retry of the total serial data transfer can then commence.

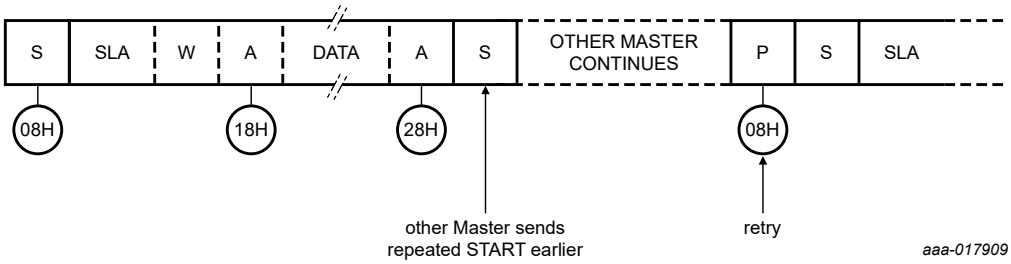


Figure 44. Simultaneous Repeated-START conditions from two masters

19.10.6.2 Data transfer after loss of arbitration

Arbitration may be lost in the master transmitter and master receiver modes (see Figure 38). The following states in STAT indicate loss of arbitration: 0x38, 0x68, 0x78, and 0xB0 (see Figure 40 and Figure 41).

If the routines which service these states set the STA flag in CON, then, if the bus is free again, a START condition (state 0x08) is transmitted.

19.10.6.3 Forced access to the I^2C-bus

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, interference, temporary interruption of the bus, or a temporary short-circuit between SDA and SCL may cause this problem.

If an uncontrolled source generates a superfluous START or masks a STOP condition, the I^2C-bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable time, a forced access to the I^2C-bus is possible. This forced access is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The I^2C-bus hardware behaves as if a STOP condition was received and is able to transmit a START condition. Hardware clears the STO flag (see Figure 45).

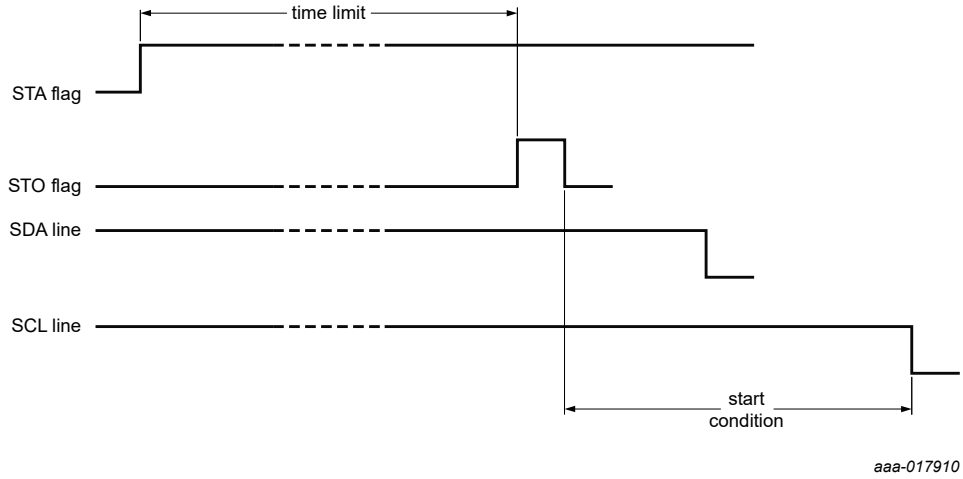
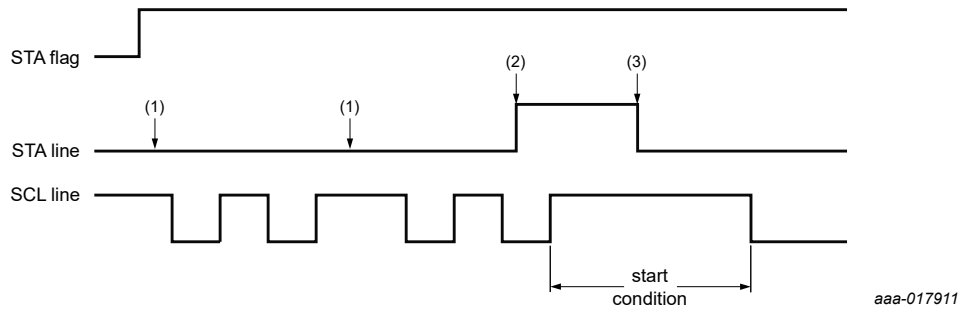


Figure 45. Forced access to a busy I2C-bus

19.10.6.4 I²C-bus obstructed by a low level on SCL or SDA

If either the SDA or SCL line is held low by any device on the bus, an I²C-bus hang-up can occur. If the SCL line is obstructed (pulled low) by a device on the bus, no further serial transfer is possible. The device that is pulling the SCL bus line low must resolve this problem.

Typically, another device on the bus, that has become out of synchronization with the current bus master, may obstruct the SDA line. This loss of synchronization may occur by either missing a clock or by sensing a noise pulse as a clock. In this case, the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 46). The I²C-bus interface does not include a dedicated timeout timer to detect an obstructed bus. However, such a timer can be implemented using another timer in the system. When detected, software can force clocks (up to 9 may be required) on SCL until the offending device releases SDA. At that point, the slave may still be out of synchronization, so, to ensure that all I²C-bus peripherals are synchronized, a START must be generated.



1. Unsuccessful attempt to send a START condition.
2. SDA line is released.
3. Successful attempt to send a START condition. State 08h is entered.

Figure 46. Recovering from a bus obstruction caused by a LOW level on SDA

19.10.6.5 Bus error

When a START or STOP condition is detected at an illegal position in the format frame, a bus error occurs. Examples of illegal positions are during the serial transfer of an address byte, a data bit, or an acknowledge bit.

The I^2C-bus hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, the I^2C-bus block immediately switches to the not addressed slave mode. It then releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 0x00. This status code may be used to vector to a state service routine which either attempts the aborted serial transfer again, or simply recovers from the error condition as shown in Table 227.

19.10.7 I^2C-bus state service routines

This section provides examples of operations that the various I^2C-bus state service routines must. These operations include:

- Initialization of the I^2C-bus block after a reset
- I^2C-bus interrupt service
- The 26 state service routines support all four I^2C-bus operating modes

19.10.8 Initialization

In the initialization example, the I^2C-bus block is enabled for master and slave modes. For each mode, a buffer is used for transmission and reception. The initialization routine performs the following functions:

- The slave address of the part, and the general call bit (GC), are loaded into ADR.
- The I^2C-bus interrupt enable and interrupt priority bits are set.
- Simultaneously setting the I2EN and AA bits in CON enables the slave mode.
- Loading the SCLH and SCLL registers defines the serial clock frequency (for master modes). The master routines must be started in the main program.

The I^2C-bus hardware now begins checking the I^2C-bus for its own slave address and general call. If the general call or the own slave address of the I^2C-bus block is detected, an interrupt is requested and STAT is loaded with the appropriate state information.

19.10.9 I^2C-bus interrupt service

When the I^2C-bus interrupt is entered, STAT contains a status code which identifies one of the 26 state services to be executed.

19.10.10 The state service routines

Each state routine is part of the I^2C-bus interrupt routine and handles one of the 26 states.

19.10.11 Adapting state services to an application

The state service examples show the typical actions that must be performed in response to the 26 I^2C-bus state codes. If one or more of the four I^2C-bus operating modes are not used, the associated state services can be omitted, as long as care is taken that those states can never occur.

In an application, it may be desirable to implement a timeout during I^2C-bus operations to trap an inoperative bus or a lost service routine.

20 Serial wire debug (SWD)

20.1 About this chapter

This chapter describes the features of the implemented serial wire debug (SWD).

20.2 Introduction

SWD provides a clock (SWCLK) plus a single bidirectional data (SWDIO) connection. This data connection provides all normal JTAG debug and test functionality. It also provides real-time access to system memory, without halting the processor or requiring any target resident code. To pass data between the debugger and the target system in a highly efficient and standard way, SWD uses an Arm standard bidirectional wire protocol. This protocol is defined in Arm Debug Interface v5.

20.3 Features

The Arm Cortex-M0+ is configured to support up to four breakpoints and two watchpoints.

- Supports Arm SWD mode
- Direct debug access to all memories, registers, and peripherals
- No target resources are required for the debugging session
- Four breakpoints
  Four instruction breakpoints that can also be used to remap instruction addresses for code patches. Two data comparators that can be used to remap addresses for patches to literal values.
- Two data watchpoints that can also be used as triggers

Resetting the chip does not reset the debug access point.

Note: *If a debug session is in progress, do not reset the chip.*

21 Serial peripheral interface (SPI)

21.1 About this chapter

This chapter describes the serial peripheral interface (SPI; APB 0x4004 0000).

21.2 Features

- Compatible with Motorola SPI, 4-wire TI SSI, and National Semiconductor Microwire buses
- Synchronous serial communication
- Supports master or slave operation
- Eight-frame FIFOs for both transmit and receive
- 4-bit to 16-bit frame

21.3 General description

The SPI/SSP is a synchronous serial port (SSP) controller capable of operation on an SPI, 4-wire SSI, or Microwire bus. It can interact with multiple masters and slaves on the bus. Only a single master and a single slave can communicate on the bus during a given data transfer. Data transfers are in principle full duplex, with frames of 4 bits to 16 bits of bidirectional data flowing between master and slave. In practice, often only one of these two data flows carries meaningful data.

21.4 Configuration

The serial peripheral interface is configured using the following registers:

- Pins:
  The SPI pins must be configured in the IOCON register bock. In addition, use the IOCON_LOC register to select a location for the SCLK function.
- Power:
  In the SYSAHBCLKCTRL register, set bit 7 and bit 18.
- Peripheral clock:
  Enable the SPI peripheral clock by writing to the SSPCLKDIV registers.
- Reset:
  Before accessing the SPI blocks, ensure that the SSP_RST_N bits (bit 0 and bit 2) in the PRESETCTRL register (Table 12) are set to 1. This procedure deasserts the reset signal to the SPI blocks.

21.5 Pin description

Table 228. SPI pin description

Pin name	Type	Interface pin SPI	SSI	Microwire	Description
SCLK	I/O	SCLK	CLK	SK	serial clock
SSEL	I/O	SSEL	FS	CS	frame sync/slave select
MISO	I/O	MISO	DR (M) DX (S)	SI (M) SO (S)	master in slave out

Table 228. SPI pin description...*continued*

Pin name	Type	Interface pin SPI	SSI	Microwire	Description
MOSI	I/O	MOSI	DX (M) DR (S)	SO (M) SI (S)	master out slave in

21.5.1 Pin detailed description

Serial clock

SCK/CLK/SK is a clock signal used to synchronize the transfer of data. The master drives the clock signal and the slave receives it. When SPI/SSP interface is used, the clock is programmable to be active-HIGH or active-LOW. Otherwise, it is always active-HIGH. SCK only switches during a data transfer. At any other time, the SPI/SSP interface either stays in its inactive state or is not driven (remains in high-impedance state).

Frame sync/slave select

When the SPI/SSP interface is a bus master, it drives this signal to an active state before the start of serial data. It then releases it to an inactive state after the data has been sent. The active state can be HIGH or LOW depending upon the selected bus and mode. When the SPI/SSP interface is a bus slave, this signal qualifies the presence of data from the master according to the protocol in use.

When there is only one master and slave, the frame sync or slave select signal from the master can be connected directly to the corresponding input of the slave. When there are multiple slaves, further qualification of frame select/slave select inputs is normally necessary to prevent more than one slave from responding to a transfer.

Master in slave out

The MISO signal transfers serial data from the slave to the master. When the SPI/SSP is a slave, it outputs serial data on this signal. When the SPI/SSP is a master, it clocks in serial data from this signal. It does not drive this signal and leaves it in a high-impedance state when the SPI/SSP is a slave and not selected by FS/SSEL.

Master out slave in

The MOSI signal transfers serial data from the master to the slave. When the SPI/SSP is a master, it outputs serial data on this signal. When the SPI/SSP is a slave, it clocks in serial data from this signal.

21.6 Functional description

21.6.1 Texas Instruments synchronous serial frame format

For a device configured as a master in this mode (see Figure 47), CLK and FS are forced LOW. The transmit data line DX is in 3-state mode whenever the SSP is idle. When the bottom entry of the transmit FIFO contains data, FS is pulsed HIGH for one CLK period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of CLK, the MSB of the 4-bit to 16-bit data frame is shifted out on the DX pin. Likewise, the MSB of the received data is shifted onto the DR pin by the off-chip serial slave device.

The SSP and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each CLK. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of CLK after the LSB has been latched.

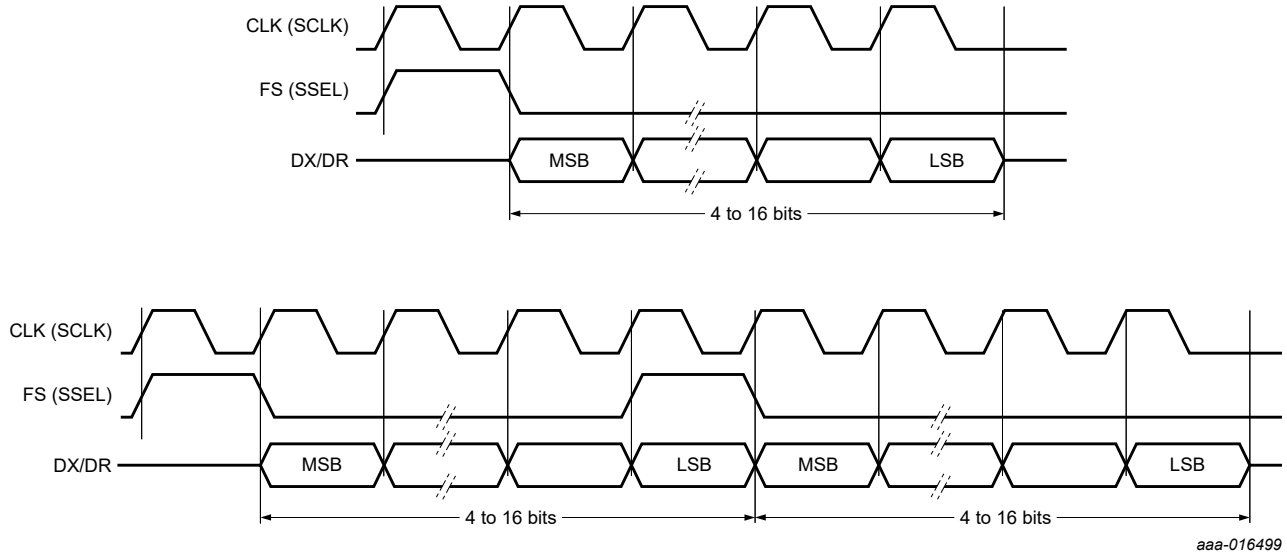


Figure 47. Texas Instruments synchronous Serial Frame Format: Single frame transfer (top), back-to-back two frames transfer (bottom)

21.6.2 SPI frame format

The SPI interface is a four-wire interface where the SSEL signal behaves as a slave select. The main feature of the SPI format is that the inactive state and phase of the SCK signal are programmable. This programming is done through the CPOL and CPHA bits within the SSPCR0 control register.

21.6.2.1 Clock Polarity (CPOL) and phase (CPHA) control

When the CPOL clock polarity control bit is LOW, it produces a steady state LOW value on the SCK pin. If the CPOL clock polarity control bit is HIGH, a steady state HIGH value is placed on the CLK pin when data is not transferred.

The CPHA control bit selects the clock edge that captures data and allows it to change state. It has the greatest impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. When the CPHA phase control bit is LOW, data is captured on the first clock edge transition. If the CPHA clock phase control bit is HIGH, data is captured on the second clock edge transition.

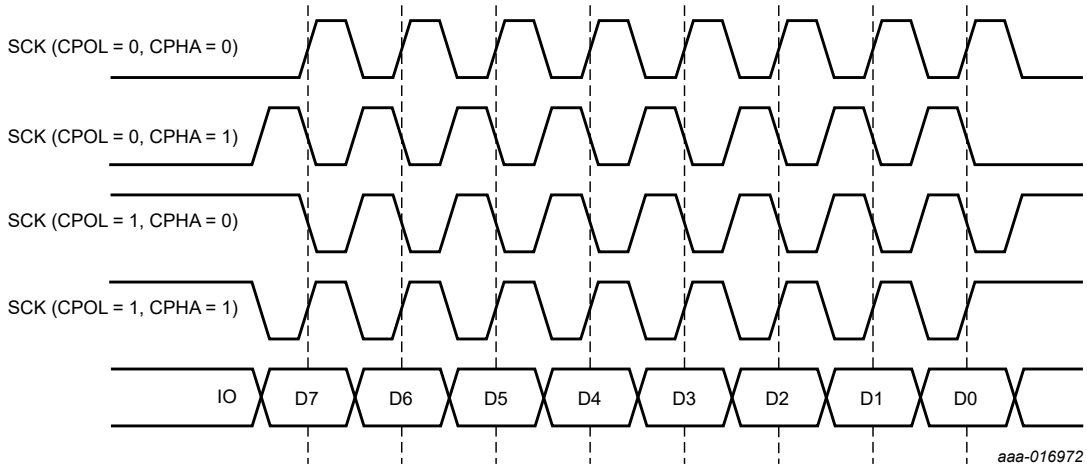


Figure 48. SPI bus timing

21.6.2.2 SPI format with CPOL = logic 0, CPHA = logic 0

Figure 49 shows single and continuous transmission signal sequences for SPI format with CPOL = logic 0, CPHA = logic 0.

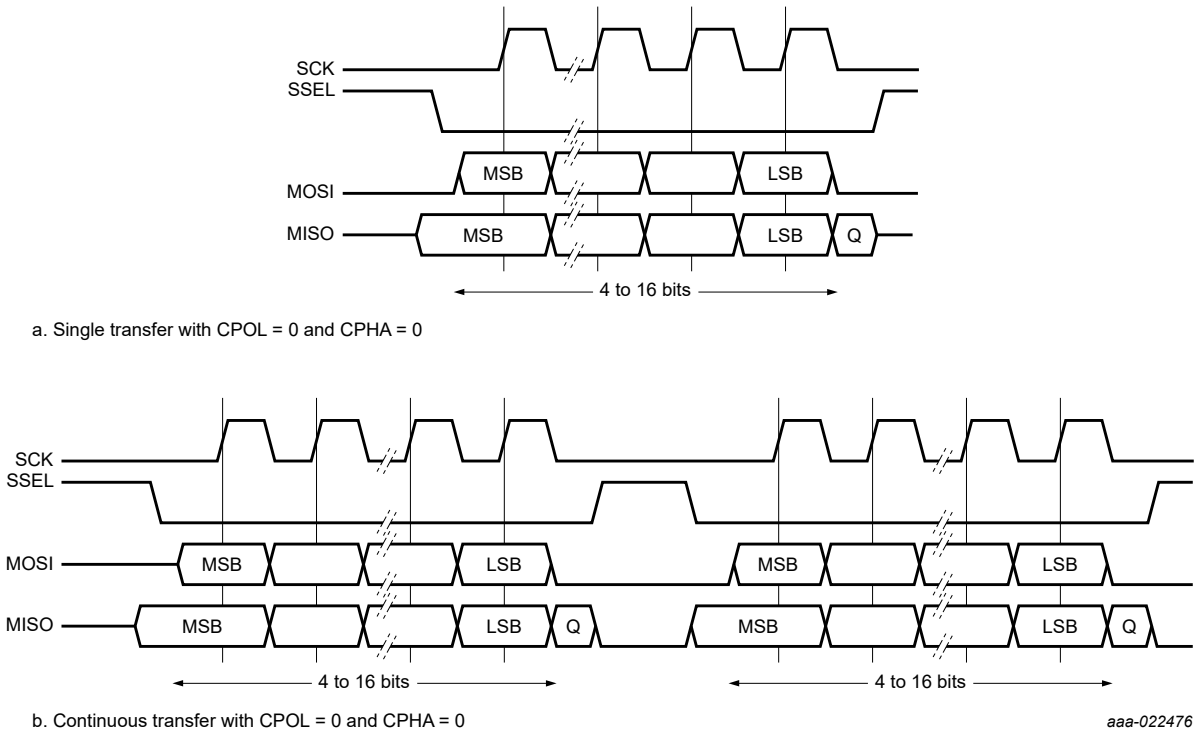


Figure 49. SPI frame format with CPOL = logic 0 and CPHA = logic 0. (a) Single and b) Continuous Transfer

In this configuration, during idle periods:

- The CLK signal is forced LOW
- SSEL is forced HIGH
- The transmit MOSI/MISO pad is in high impedance

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the SSEL master signal driven LOW signifies the start of transmission. This signal causes slave data to be enabled onto the MISO input line of the master. The MOSI of the master is enabled.

One half SCK period later, valid master data is transferred to the MOSI pin. Now that the master and slave data have been set, the SCK master clock pin goes HIGH after one further half SCK period.

The data is captured on the rising and propagated on the falling edges of the SCK signal.

For a single word transmission, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSEL signal must be pulsed HIGH between each data word transfer. The reason is that if the CPHA bit is logic 0, the data in the serial peripheral register cannot be altered. So, the master device must raise the SSEL pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSEL pin is returned to its idle state one SCK period after the last bit has been captured.

21.6.2.3 SPI format with CPOL = logic 0, CPHA = logic 1

Figure 50 shows the transfer signal sequence for SPI format with CPOL = logic 0, CPHA = logic 1, which covers single and continuous transfers.

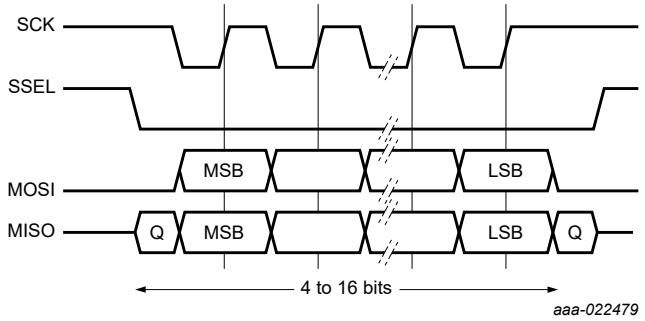


Figure 50. SPI frame format with CPOL = logic 0 and CPHA = logic 1

In this configuration, during idle periods:

- The CLK signal is forced LOW
- SSEL is forced HIGH
- The transmit MOSI/MISO pad is in high impedance

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the SSEL master signal driven LOW signifies the start of transmission. The MOSI pin of the master is enabled. After a further one half SCK period, master and slave valid data is enabled onto their respective transmission lines. At the same time, the SCK is enabled with a rising edge transition.

Data is then captured on the falling edges and propagated on the rising edges of the SCK signal. After transferring a single word, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured. For continuous back-to-back transfers, the SSEL pin is held LOW between successive data words. Termination is the same as for the single word transfer.

21.6.2.4 SPI format with CPOL = logic 1, CPHA = logic 0

Figure 51 shows single and continuous transmission signal sequences for SPI format with CPOL = logic 1, CPHA = logic 0.

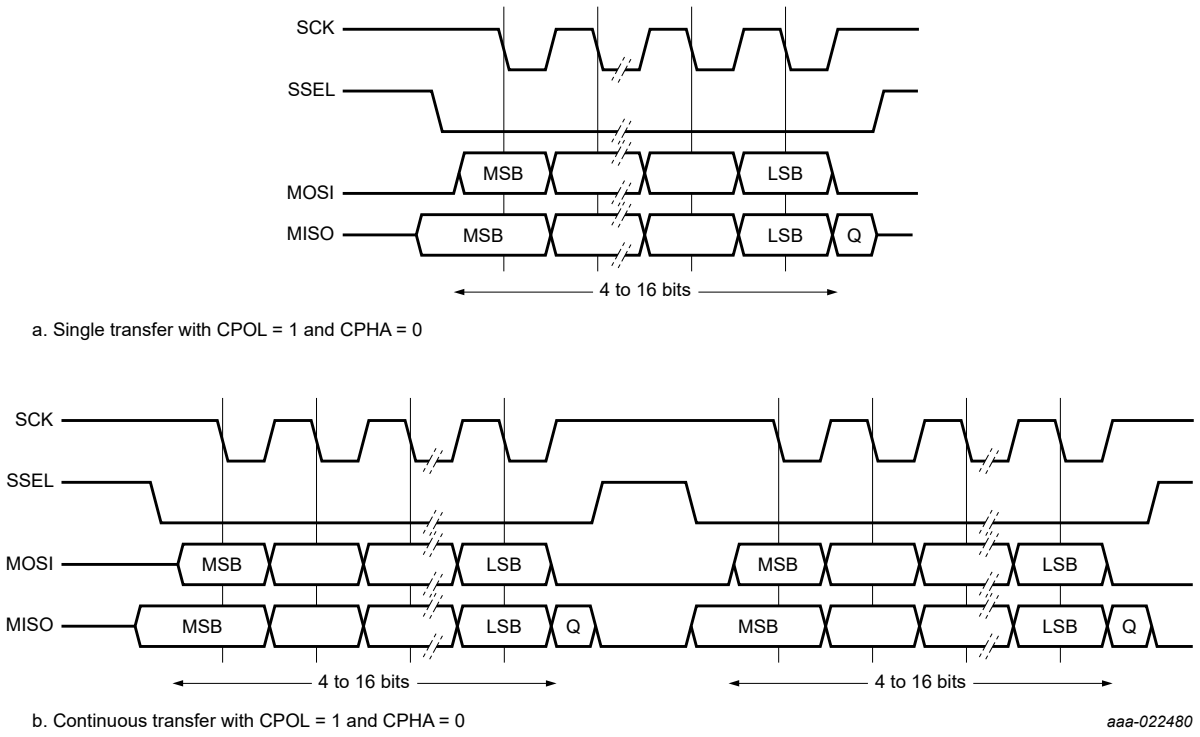


Figure 51. SPI frame format with CPOL = logic 1 and CPHA = logic 0 (a) Single and b) Continuous Transfer)

In this configuration, during idle periods:

- The CLK signal is forced HIGH
- SSEL is forced HIGH
- The transmit MOSI/MISO pad is in high impedance

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the SSEL master signal driven LOW signifies the start of transmission. This signal causes slave data to be transferred immediately onto the MISO line of the master. The MOSI pin of the master is enabled.

One half period later, valid master data is transferred to the MOSI line. Now that the master and slave data have been set, the SCK master clock pin becomes LOW after one further half SCK period. So, data is captured on the falling edges and is propagated on the rising edges of the SCK signal.

For a single word transmission, after all bits have been transferred, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

However, in case of continuous back-to-back transmissions, the SSEL signal must be pulsed HIGH between each data word transfer. The reason is that if the CPHA bit is logic 0, the data in the serial peripheral register cannot be altered. So, to enable the serial peripheral data write, the master device must raise the SSEL pin of the slave device between each data transfer. On completion of the continuous transfer, the SSEL pin is returned to its idle state one SCK period after the last bit has been captured.

21.6.2.5 SPI format with CPOL = logic 1, CPHA = logic 1

Figure 52 shows the transfer signal sequence for SPI format with CPOL = logic 1, CPHA = logic 1, which covers both single and continuous transfers.

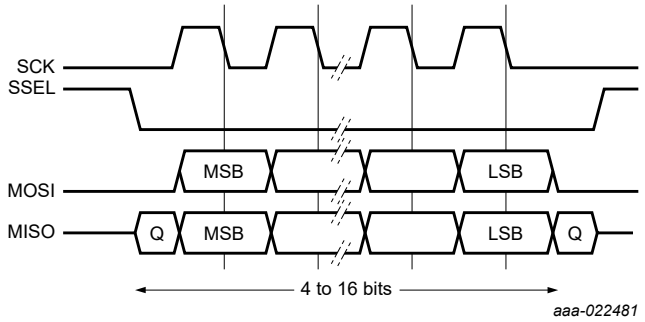


Figure 52. SPI frame format with CPOL = logic 1 and CPHA = logic 1

In this configuration, during idle periods:

- The CLK signal is forced HIGH
- SSEL is forced HIGH
- The transmit MOSI/MISO pad is in high impedance

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the SSEL master signal driven LOW signifies the start of transmission. The MOSI pin of the master is enabled. After a further one half SCK period, master and slave data are enabled into their respective transmission lines. At the same time, the SCK is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SCK signal.

In a single word transmission, after all bits have been transferred, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

For continuous back-to-back transmissions, the SSEL pins remain in their active LOW state until the final bit of the last word has been captured. They then return to their idle state as described above.

Generally, for continuous back-to-back transfers, the SSEL pin is held LOW between successive data words. Termination is the same as for a single word transfer.

21.6.3 Semiconductor Microwire frame format

Figure 53 shows the Microwire frame format for a single frame. Figure 54 shows the same format when back-to-back frames are transmitted.

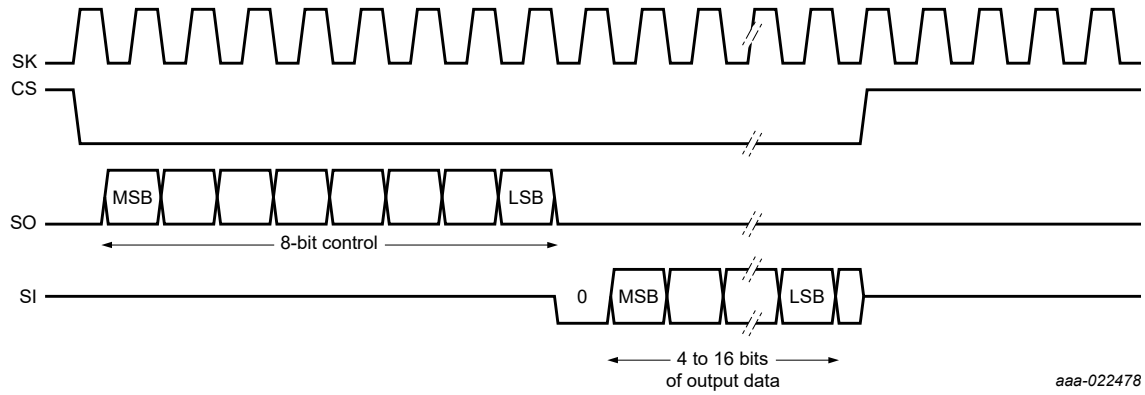


Figure 53. Microwire frame format (single transfer)

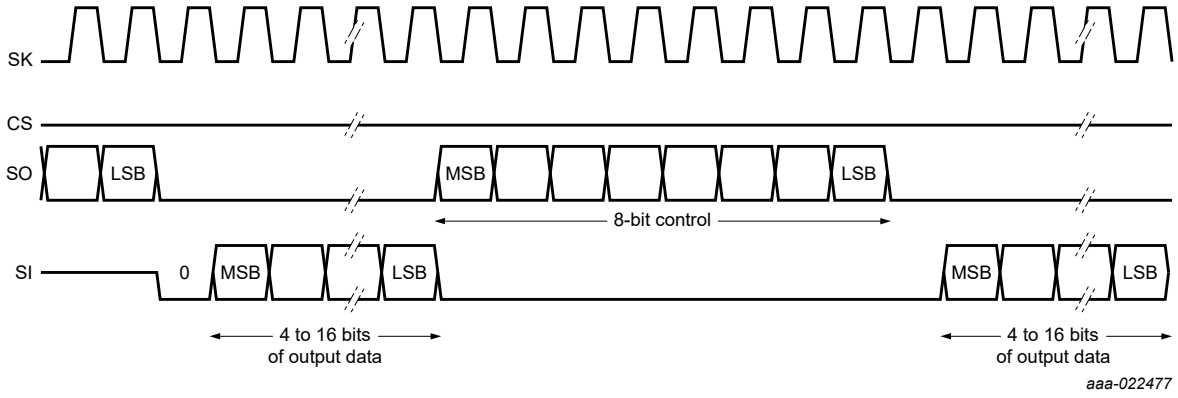


Figure 54. Microwire frame format (continuous transfers)

Microwire format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SPI/SSP to the off-chip slave device. The SPI/SSP does not receive any incoming data during this transmission. After the message has been sent, the off-chip slave decodes it. Then, after waiting one serial clock after the last bit of the 8-bit control message has been sent, it responds with the required data. The returned data is 4 bits to 16 bits in length, making the total frame length anywhere from 13 bits to 25 bits.

In this configuration, during idle periods:

- The SK signal is forced LOW
- CS is forced HIGH
- The transmit data line SO is arbitrarily forced LOW

Writing a control byte to the transmit FIFO triggers a transmission. The falling edge of CS causes the bottom entry value of the transmit FIFO to be transferred to the serial shift register of the transmit logic. It also causes the MSB of the 8-bit control frame to be shifted out onto the SO pin. CS remains LOW during the frame transmission. The SI

pin remains in 3-state during this transmission. The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SK. After the slave device latches the last bit, the control byte is decoded during a one clock wait state. The slave responds by transmitting data back to the SPI/SSP. Each bit is driven onto SI line on the falling edge of SK. The SPI/SSP in turn latches each bit on the rising edge of SK. At the end of the frame, for single transfers, the CS signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter. This signal causes the data to be transferred to the receive FIFO.

Note: *The off-chip slave device can 3-state the receive line either on the falling edge of SK after the receive shifter latches the LSB or when the CS pin goes HIGH.*

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the CS line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each received value is transferred from the receive shifter on the falling edge SK, after the LSB of the frame has been latched into the SPI/SSP.

21.7 Register descriptions

The register addresses of the SPI controllers are shown in Table 229.

The reset values only reflect the data stored in used bits. It does not include the content of reserved bits.

Table 229. Register overview: Serial peripheral interface (base address 0x4004 0000)

Name	Access	Address offset	Description	Reset value
CR0	RW	0x000	Control register logic 0. Selects the serial clock rate, bus type, and data size	0x0000 000F
CR1	RW	0x004	Control register logic 1. Selects master/slave and other modes	0x0000 0000
DR	RW	0x008	Data register. Writes fill the transmit FIFO, and reads empty the receive FIFO	0x0000 0000
SR	R	0x00C	Status register	0x0000 0003
CPSR	RW	0x010	Clock prescale register	0x0000 0000
IMSC	RW	0x014	Interrupt mask set and clear register	0x0000 0000
RIS	R	0x018	Raw interrupt status register	0x0000 0008
MIS	R	0x01C	Masked interrupt status register	0x0000 0000
ICR	W	0x020	Interrupt clear register	n/a

21.7.1 SPI/SSP control register 0 (CR0)

This register controls the basic operation of the SPI/SSP controller.

Always write settings to this register, even if the desired settings match the reset default.

Table 230. CR0 register (address 0x4004 0000) bit description

Bit	Symbol	Description		Reset value
3:0	DSS	Data size select. This field controls the number of bits transferred in each frame. Do not use values 0000-0010 as they are not supported.		0Fh
		3h	4-bit transfer	
		4h	5-bit transfer	
		5h	6-bit transfer	
		6h	7-bit transfer	
		7h	8-bit transfer	
		8h	9-bit transfer	
		9h	10-bit transfer	
		Ah	11-bit transfer	
		Bh	12-bit transfer	
		Ch	13-bit transfer	
		Dh	14-bit transfer	
		Eh	15-bit transfer	
		Fh	16-bit transfer	
5:4	FRF	Frame format		0
		00b	SPI	
		01b	TI	
		10b	Microwire	
		11b	(This combination is not supported. Do not use it.)	
6	CPOL	Clock out polarity. This bit is only used in SPI mode.		0
		0	SPI controller maintains the bus clock LOW between frames.	
		1	SPI controller maintains the bus clock HIGH between frames.	
7	CPHA	Clock out phase. This bit is only used in SPI mode.		0
		0	SPI controller captures serial data on the first clock transition of the frame, that is, the transition away from the inter-frame state of the clock line.	
		1	SPI controller captures serial data on the second clock transition of the frame, that is, the transition back to the inter-frame state of the clock line.	
15:8	SCR	Serial clock rate. The number of prescaler output clocks per bit on the bus, minus one. Given that CPSDVSR is the prescale divider and the APB clock PCLK clocks the prescaler, the bit frequency is PCLK / (CPSDVSR × [SCR + 1])		0
31:16	-	reserved		-

21.7.2 SPI/SSP control register 1 (CR1)

This register controls the certain aspects of the SPI/SSP controller.

Table 231. CR1 register (address 0x4004 0004) bit description

Bit	Symbol	Description		Reset value
0	LBM	Loop back mode		0
		0	During normal operation	
		1	Serial input is taken from the serial output (MOSI or MISO) rather than the serial input pin (MISO or MOSI respectively)	
1	SSE	SPI enable		0
		0	The SPI controller is disabled.	
		1	The SPI controller interacts with other devices on the serial bus. Before setting this bit, software must write the appropriate control information to the other SPI/SSP registers and interrupt controller registers.	
2	MS	Master/slave mode. This bit can only be written when the SSE bit is logic 0.		0
		0	The SPI controller acts as a master on the bus, driving the SCLK, MOSI, and SSEL lines and receiving the MISO line.	
		1	The SPI controller acts as a slave on the bus, driving MISO line and receiving SCLK, MOSI, and SSEL lines.	
3	SOD	Slave output disable. This bit is relevant only in slave mode (MS = logic 1). If it is logic 1, this SPI controller is blocked from driving the transmit data line (MISO).		0
31:4	-	reserved		-

21.7.3 SPI/SSP data register (DR)

Software can write data to be transmitted to this register and read data that has been received.

Table 232. DR register 0 (address 0x4004 0008) bit description

Bit	Symbol	Value	Description	Reset value
15:0	DATA	-	Write: Whenever the TNF bit in the status register is logic 1, software can write data to be sent in a future frame to this register. This value indicates that the TX FIFO is not full. If the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data begins immediately. Otherwise, the data written to this register is sent as soon as all previous data has been sent (and received). If the data length is less than 16 bits, software must right-justify the data written to this register. Read: Whenever the RNE bit in the Status register is logic 1, software can read data from this register. This value indicates that the RX FIFO is not empty. When software reads this register, the SPI controller returns data from the least recent frame in the RX FIFO. If the data length is less than 16 bits, the data is right-justified in this field with higher-order bits filled with logic 0s.	0
31:16	-	-	reserved	-

21.7.4 SPI/SSP status register (SR)

This read-only register reflects the status of the SPI controller.

Table 233. SR register (address 0x4004 000C) bit description

Bit	Symbol	Value	Description	Reset value
0	TFE	-	Transmit FIFO empty. This bit is logic 1 if the Transmit FIFO is empty, logic 0 if not.	1
1	TNF	-	Transmit FIFO not full. This bit is logic 0 if the TX FIFO is full, logic 1 if not.	1
2	RNE	-	Receive FIFO Not Empty. This bit is logic 0 if the RX FIFO is empty, logic 1 if not.	0
3	RFF	-	Receive FIFO Full. This bit is logic 1 if the RX FIFO is full, logic 0 if not.	0
4	BSY	-	Busy. This bit is logic 0 if the SPI controller is idle, logic 1 if it is sending/ receiving a frame and/or the TX FIFO is not empty.	0
31:5	-	-	reserved	-

21.7.5 SPI/SSP clock prescale register (CPSR)

This register controls the factor by which the prescaler divides the SPI peripheral clock SPI_PCLK to yield the prescaler clock. The prescaler clock is, in turn, divided by the SCR factor in the SSPCR0 registers to determine the bit clock.

Table 234. CPSR register (address 0x4004 0010) bit description

Bit	Symbol	Value	Description	Reset value
7:0	CPSDVSR	-	An even value between 2 and 254, by which SPI_PCLK is divided to yield the prescaler output clock. Bit 0 always reads as logic 0.	0
31:8	-	-	reserved	-

21.7.6 SPI/SSP interrupt mask set/clear register (IMSC)

This register controls if each of the four possible interrupt conditions in the SPI controller is enabled.

Note: *Arm uses the word 'masked' in the opposite sense from classic computer terminology, in which 'masked' meant 'disabled'. Arm uses the word 'masked' to mean 'enabled'. To avoid confusion, we avoid using the word 'masked'.*

Table 235. IMSC register (address 0x4004 0014) bit description

Bit	Symbol	Value	Description	Reset value
0	RORIM	-	When a receive overrun occurs, software must set this bit to enable interrupt, that is when the RX FIFO is full and another frame is received. The Arm specification implies that, in this instance, the new frame data overwrites the preceding frame data.	0
1	RTIM	-	When a receive timeout condition occurs, software must set this bit to enable interrupt. When the RX FIFO is not empty and no has not been read for a timeout period, a receive timeout occurs. The SSP bit rate determines the timeout period, which is the same for master and slave modes: 32 bits at PCLK / (CPSDVSR [SCR+1]).	0
2	RXIM	-	When the RX FIFO is at least half full, software must set this bit to enable interrupt.	0

Table 235. IMSC register (address 0x4004 0014) bit description*...continued*

Bit	Symbol	Value	Description	Reset value
3	TXIM	-	When the TX FIFO is at least half empty, software must set this bit to enable interrupt.	0
31:54	-	-	reserved	-

21.7.7 SPI/SSP raw interrupt status register (RIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted, regardless of the interrupt being enabled in the SSPIMSC registers.

Table 236. RIS register (address 0x4004 0018) bit description

Bit	Symbol	Value	Description	Reset value
0	RORRIS	-	If another frame was received while the RX FIFO was full, this bit is logic 1 l. The Arm specification implies that, in this case, the new frame data overwrites the preceding frame data.	0
1	RTRIS	-	If the RX FIFO is not empty and has not been read for a timeout period, this bit is logic 1. The SSP bit rate determines the timeout period, which is the same for master and slave modes: 32 bits at PCLK / (CPSDVSR [SCR+1])	0
2	RXRIS	-	If the RX FIFO is at least half full, this bit is logic 1.	0
3	TXRIS	-	If the TX FIFO is at least half empty, this bit is logic 1.	1
31:4	-	-	reserved	-

21.7.8 SPI/SSP masked interrupt status register (MIS)

This read-only register contains a logic 1 for each interrupt condition that is asserted and enabled in the SSPIMSC registers. When an SPI interrupt occurs, the interrupt handler must read this register to determine the cause or causes of the interrupt.

Table 237. MIS register (address 0x4004 001C) bit description

Bit	Symbol	Value	Description	Reset value
0	RORMIS	-	If another frame was received while the RX FIFO was full and this interrupt is enabled, this bit is logic 1.	0
1	RTMIS	-	If the RX FIFO is not empty, has not been read for a timeout period, and this interrupt is enabled, this bit is logic 1. The SSP bit rate determines the timeout period, which is the same for master and slave modes: 32 bits at PCLK / (CPSDVSR [SCR+1])	0
2	RXMIS	-	If the RX FIFO is at least half full and this interrupt is enabled, this bit is logic 1.	0
3	TXMIS	-	If the TX FIFO is at least half empty and this interrupt is enabled, this bit is logic 1.	0
31:4	-	-	reserved	-

21.7.9 SPI/SSP interrupt clear register (ICR)

To clear the corresponding interrupt condition or conditions in the SPI controller, software can write one or more logic 1s to this write-only register.

Note: *The other two interrupt conditions can be cleared by writing or reading the appropriate FIFO or disabled by clearing the corresponding bit in SSPIMSC registers.*

Table 238. ICR register (address 0x4004 0020) bit description

Bit	Symbol	Value	Description	Reset value
0	RORIC	-	Writing a logic 1 to this bit clears the ’frame was received when RX FIFO was full’ interrupt.	-
1	RTIC	-	Writing a logic 1 to this bit clears the RX FIFO if it has not been read for a timeout period interrupt. The SSP bit rate determines the timeout period, which is the same for master and slave modes: 32 bits at PCLK / (CPSDVSR [SCR +1]).	-
31:2	-	-	reserved	-

22 General-purpose input output (GPIO)

22.1 About this chapter

This chapter describes the GPIO (AHB 0x5000 0000).

22.2 Features

- Software can configure GPIO pins as input or output
- Each individual port pin can serve as an edge or level-sensitive interrupt request
- Interrupts can be configured on single falling or rising edges and on both edges
- Level-sensitive interrupt pins can be HIGH or LOW-active
- All GPIO pins are inputs by default
- Address bits 13:2 mask the reading and writing of GPIODATA registers

22.3 General description

Each GPIO register can be up to 12 bits wide and can be read or written using word or halfword operations at word addresses.

The reset values only reflect the data stored in used bits. It does not include the content of reserved bits.

22.4 Functional description

22.4.1 Write/read data operation

Bits [13:2] of a 14-bit wide address bus are used to create a 12-bit wide mask for write and read operations on the 12 GPIO pins for each port. This mask enables software to set GPIO bits without affecting any other pins in a single write operation. Read and write operations only affect GPIODATA bits masked by logic 1.

The masked GPIODATA register can be located anywhere between address offsets 0x0000 and 0x3FFC in the GPIO address space. Reading from and writing to the GPIODATA register at address 0x3FFC sets all masking bits to logic 1.

22.4.1.1 Write operation

If address bit (i + 2) associated with the GPIO port bit i (i = 0 to 11) to be written is HIGH, the GPIODATA register bit i is updated. If the address bit (i + 2) is LOW, the corresponding GPIODATA register bit i is left unchanged.

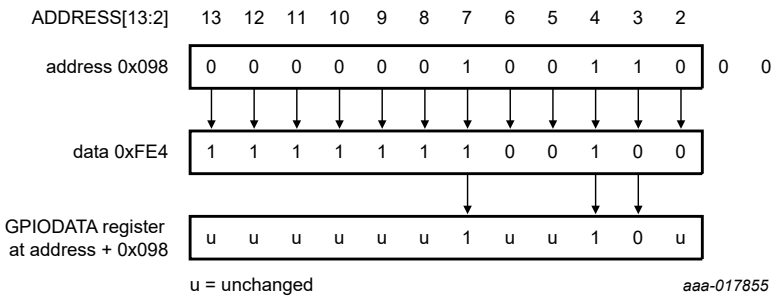


Figure 55. Masked write operation to the GPIODATA register

22.4.1.2 Read operation

If the address bit associated with the GPIO data bit is HIGH, the value is read. If the address bit is LOW, the GPIO data bit is read as logic 0. Reading a port DATA register yields the state of port pins 11:0 ANDed with address bits 13:2.

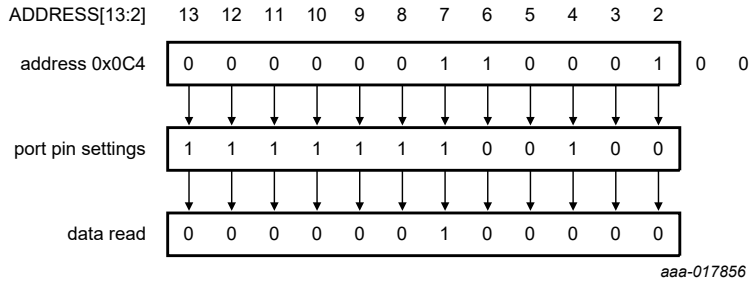


Figure 56. Masked GPIO read operation

22.5 Register descriptions

Table 239. Register overview: General-purpose I/O (base address 0x5000 0000)

Name	Access	Address offset	Description	Reset value
GPIODATA	RW	0x0000 - 0x3FF8	Data address masking register locations for pins PIO0_0 to PIO0_11.	n/a
GPIODATA	RW	0x3FFC	Data register for pins PIO_0 to PIO0_11.	n/a
-	-	0x4000 - 0x7FFC	reserved	-
DIR	RW	0x8000	Data direction register	0x0000 0000
IS	RW	0x8004	Interrupt sense register	0x0000 0000
IBE	RW	0x8008	Interrupt both edges register	0x0000 0000
IEV	RW	0x800C	Interrupt event register	0x0000 0000
IMSC	RW	0x8010	Interrupt mask register	0x0000 0000
RIS	R	0x8014	Raw interrupt status register	0x0000 0000
MIS	R	0x8018	Masked interrupt status register	0x0000 0000
IC	W	0x801C	Interrupt clear register	0x0000 0000
-	-	0x8020 - 0xFFFF	reserved	-

22.5.1 GPIO data register (GPIODATA)

The GPIODATA register holds the current logic state of the pin, independently of it being configured as a GPIO input or output or as another other digital function. If the pin is configured as GPIO output, the current value of the GPIODATA register is driven to the pin.

Table 240. GPIODATA register (address 0x5000 0000 to 0x5000 3FFC) bit description

Bit	Symbol	Description	Reset value	Access
11:0	DATA	Logic levels for pins PIO0_0 to PIO0_11. HIGH = logic 1, LOW = logic 0.	n/a	RW
31:12	-	reserved	-	-

A read of the GPIODATA register always returns the current logic level (state) of the pin independently of its configuration. There is a single data register for the value of the output driver and the input state of the input of the pin. Because of this shared use of this register, write operations have different effects depending on the configuration of the pin:

- If a pin is configured as GPIO input, a write to the GPIODATA register has no effect on the pin level. A read returns the current state of the pin.
- If a pin is configured as GPIO output, the current value of GPIODATA register is driven to the pin. This value can be a result of writing to the GPIODATA register. Or it can reflect the previous state of the pin. The latter is the case if the pin is switched to GPIO output from GPIO input or another digital function. A read returns the current state of the pin.
- If a pin is configured as another digital function (input or output), a write to the GPIODATA register has no effect on the pin level. A read returns the current state of the pin even if it is configured as an output. So, by reading the GPIODATA register, the digital output or input value of a function other than GPIO on that pin can be observed.

When the pins are switched from input to output, the following rules apply:

- Pin is configured as input with a HIGH level applied:
  - Change pin to output; pin drives HIGH level
- Pin is configured as input with a LOW level applied:
  - Change pin to output; pin drives LOW level

The rules show that the pins mirror the current logic level. So, when switched from input to output, floating pins may drive an unpredictable level.

22.5.2 GPIO data direction register (DIR)

Table 241. DIR register (address 0x5000 8000) bit description

Bit	Symbol	Description	Reset value	Access
11:0	IO	Selects pin x as input or output (x = 0 to 11) logic 0 = pin PIO0_x is configured as input. logic 1 = pin PIO0_x is configured as output.	0	RW
31:12	-	reserved	-	-

22.5.3 GPIO interrupt sense register (IS)

Table 242. IS register (address 0x5000 8004) bit description

Bit	Symbol	Description	Reset value	Access
11:0	ISENSE	Selects interrupt on pin x as level or edge sensitive (x = 0 to 11) logic 0 = interrupt on pin PIO0_x is configured as edge sensitive. logic 1 = interrupt on pin PIO0_x is configured as level sensitive.	0	RW
31:12	-	reserved	0	-

22.5.4 GPIO interrupt both edges sense register (IBE)

Table 243. IBE register (address 0x5000 8008) bit description

Bit	Symbol	Description	Reset value	Access
11:0	IBE	Selects interrupt on pin x as level or edge sensitive (x = 0 to 11). logic 0 = interrupt on pin PIO0_x is controlled through register GPIOIEV. logic 1 = both edges on pin PIO0_x trigger an interrupt.	0	RW
31:12	-	reserved	0	-

22.5.5 GPIO interrupt event register (IEV)

Table 244. IEV register (address 0x5000 800C) bit description

Bit	Symbol	Description	Reset value	Access
11:0	IEV	Selects interrupt on pin x to be triggered rising or falling edges (x = 0 to 11). logic 0 = depending on settings in the IS register, falling edges or LOW level on pin PIO0_x trigger an interrupt. logic 1 = depending on settings in the IS register, rising edges or HIGH level on pin PIO0_x trigger an interrupt.	0	RW
31:12	-	reserved	0	-

22.5.6 GPIO interrupt mask register (IMSC)

Bits set to HIGH in the IMSC register allow the corresponding pins to trigger their individual interrupts and the combined GPIO interrupt line. Clearing a bit disables interrupt triggering on that pin.

Table 245. IMSC register (address 0x5000 8010) bit description

Bit	Symbol	Description	Reset value	Access
11:0	MASK	Selects interrupt on pin x to be masked (x = 0 to 11). logic 0 = interrupt on pin PIO0_x is masked. logic 1 = interrupt on pin PIO0_x is not masked.	0	RW
31:12	-	reserved	0	-

22.5.7 GPIO raw interrupt status register (RIS)

Bits read HIGH in the RIS register reflect the raw (prior to masking) interrupt status of the corresponding pins. This status indicates that, before they are allowed to trigger the IMSC, all the requirements have been met. Bits read as zero indicate that the corresponding input pins have not initiated an interrupt. The register is read-only.

Table 246. RIS register (address 0x5000 8014) bit description

Bit	Symbol	Description	Reset value	Access
11:0	RAWST	Raw interrupt status (x = 0 to 11). logic 0 = no interrupt on pin PIO0_x. logic 1 = interrupt requirements met on PIO0_x.	0	RW
31:12	-	reserved	0	-

22.5.8 GPIO masked interrupt status register (MIS)

Bits read HIGH in the MIS register reflect the status of the input lines triggering an interrupt. Bits read as LOW indicate that either no interrupt on the corresponding input pins has been generated or that the interrupt is masked. MIS is the state of the interrupt after masking. The register is read-only.

Table 247. MIS register (address 0x5000 8018) bit description

Bit	Symbol	Description	Reset value	Access
11:0	MASK	Selects interrupt on pin x to be masked (x = 0 to 11). logic 0 = no interrupt or interrupt masked on pin PIO0_x. logic 1 = interrupt on PIO0_x.	0	R
31:12	-	reserved	0	-

22.5.9 GPIO interrupt clear register (IC)

This register allows software to clear edge detection for port bits that are identified as edge-sensitive in the interrupt sense register. This register has no effect on port bits identified as level sensitive.

Table 248. IC register (address 0x5000 801C) bit description

Bit	Symbol	Description	Reset value	Access
11:0	CLR	Selects interrupt on pin x to be cleared (x = 0 to 11). Clears the interrupt edge detection logic. This register is write-only. ***Note:*** *The synchronizer between the GPIO and the NVIC blocks causes a delay of 2 clocks. Two NOPs must be added after the clear of the interrupt edge detection logic before the exit of the interrupt handler.* logic 0 = no effect. logic 1 = clears edge detection logic for pin PIO0_x.	0	W
31:12	-	reserved	0	-

23 Input/output configuration

23.1 About this chapter

This chapter describes the registers forming the input/output (IO) configuration block.

23.2 General description

The I/O configuration registers control the electrical characteristics of the pads. The following features are programmable:

- Pin function
- Internal pull-up/pull-down resistor or bus keeper function
- Low-pass filter
- I^2C-bus mode for pads hosting the I^2C-bus function

The IOCON registers control the function (GPIO or peripheral function), the input mode, and the hysteresis of all PIO_m pins. In addition, the I^2C-bus pins can be configured for different I^2C-bus modes.

23.2.1 PIO0 pin functions

The FUNC bits in the IOCON registers can be set to GPIO (FUNC = 000) or to a peripheral function. If the pins are GPIO pins, the GPIO0DIR registers determine if the pin is configured as an input or output. For any peripheral function, the pin direction is controlled automatically depending on the functionality of the pins. The GPIO0DIR registers have no effect on peripheral functions.

Table 249. Pad direction and MFIO settings

Direction	A	(pad)	Note
Input	.	Z	input mode, externally driven
Output	0	0	output mode logic 0
Output	1	1	output mode logic 1

23.2.2 PIO0 pin mode

The MODE bits in the IOCON register allow the selection of on-chip pull-up or pull-down resistors for each pin or to select the repeater mode. The possible on-chip resistor configurations are pull-up enabled, pull-down enabled, or no pull-up/pull-down. The default value is all disabled. When the pin is at logic 1, the repeater mode enables the pull-up resistor. When the pin is at logic 0, it enables the pull-down resistor. This mode causes the pin to retain its last known state, if it is configured as an input and is not driven externally. The state retention is not applicable to the deep power-down mode. Repeater mode may typically be used to prevent a pin from floating when it is temporarily not driven. Allowing it to float can use significant power.

The LPF bit indicates if the low pass/glitch filtered input is selected or not.

Table 250. LPF bit and MFIO pad output selection

LPF	Pad output selected
0	ZI

Table 250. LPF bit and MFIO pad output selection*...continued*

LPF	Pad output selected
1	ZIF

23.2.3 PIO0 I^2C mode

Pads PIO0_4 and PIO0_5 can be used in I^2C-bus and GPIO mode. The pads are open-drain (in both modes). The FUNC bits of registers IOCON_PIO0_4 and IOCON_PIO0_5 select the I^2C-bus or GPIO mode. Depending on the selection, an input glitch filter is active or not:

- In I^2C-bus mode (FUNC=001b), no glitch filter is present on the input. However, there is glitch filtering in the I^2C-bus block.
- In GPIO mode (FUNC=000b), a 50 ns glitch filter is active on the input.

The I2CMODE bits of the registers IOCON_PIO0_4 and IOCON_PIO0_5 determine the output behavior, independent of the FUNC selection:

- With I2CMODE=00b (default), the selected mode is I^2C-bus standard/fast mode transmit, with turn-on slew-rate control.
- With I2CMODE=01b, the selected mode is open-drain GPIO (4 mA) without slew-rate control.

The I^2C-bus pads have no pull-up/pull down functionality.

Table 251. Pad output behavior

Requested output from IC core	I2CMODE	(pad)	Mode
0	00	0	I^2C-bus standard/fast mode transmit
1	00	Z	I^2C-bus standard/fast mode transmit
0	01	0	open-drain GPIO (4 mA)
1	01	Z	open-drain GPIO (4 mA)
0	10	0	reserved
1	10	Z	reserved
0	11	0	reserved
1	11	Z	reserved

23.2.4 PIO0 current drive mode

PIO0_3, PIO0_7, PIO0_10, and PIO0_11 are high-drive pads that can deliver up to 20 mA to the load. These PIO pins can be set to either digital mode or current sink mode. In digital mode, the output voltage of the pad switches between V_{SS} and V_{DD}. In current sink mode, the output current sink switches between $I_{drive(low)}$ and $I_{drive(high)}$ as set by the ILOW and IHI bits. The maximum pad voltage is limited to 5 V.

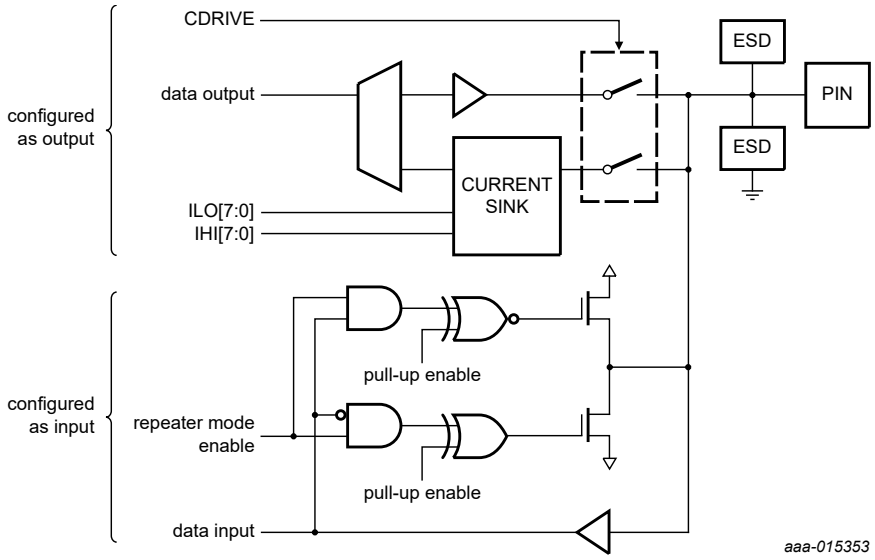


Figure 57. Pin configuration with current drive mode

The DDRIVE bit indicates if the high or the ultra-high drive is selected.

Table 252. DDRIVE bit settings

DDRIVE	Description
0	high drive (20 mA drive at 3.1 V supply)
1	ultra-high drive (20 mA drive at 2.4 V supply)

The CDRIVE bit sets the pad to the current sink mode (an analog output mode). The current drive mode can only be selected when the PIOs are in output mode. The direction of the pad is controlled via the FUNC bits and the direction register, DIR, of the GPIO block when FUNC = "000".

The IDAC controller sets the digital features of the pad to high-impedance input mode with disabled receiver. The controller directly drives the IO port of the pad according to the values of ILO and IHI.

Table 253. Pad mode as function of CDRIVE

Direction	CDRIVE	Description
Input	0	digital in
Output	0	digital out
Input	1	digital in
Output	1	current drive out

23.2.5 Reset state of pins

The reset states of the digital pins are chosen to have minimal interaction with external connections (high-Z input). All unused pins must be connected to known potential (either VDD or VSS). Do not leave connected pins floating.

The reset state of the analog pins disconnects the internal analog bus and pulls down the external pin to VSS.

23.2.6 ANA0 input selection

The analog pins have direct analog connections to the internal analog buses. ESD structures protect them. The FUNC bit in the IOCON register determines the interconnections.

23.3 IOCON register descriptions (base address 0x4004 4000)

The I/O configuration registers control the PIO port pins, inputs and outputs of all peripherals and functional blocks, I^2C-bus pins, and analog input/output pins.

Each port pin PIO0_x and ANA0_x has one IOCON register assigned to control the function and electrical characteristics of the pin.

Table 254. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
PIO0_0 [1]	RW	0x000	PIO0_0/WAKEUP	0x0000 0000	Table 255
PIO0_1	RW	0x004	PIO0_1/CLKOUT	0x0000 0000	Table 256
PIO0_2	RW	0x008	PIO0_2/SPI SSEL	0x0000 0000	Table 257
PIO0_3	RW	0x00C	PIO0_3/CT16B_M0	0x0000 0000	Table 258
PIO0_4	RW	0x010	PIO0_4/IICSCL	0x0000 0001	Table 259
PIO0_5	RW	0x014	PIO0_5/IICSDA	0x0000 0001	Table 260
PIO0_6	RW	0x018	PIO0_6/SPI SCLK	0x0000 0000	Table 261
PIO0_7	RW	0x01C	PIO0_7/CT16B_M1	0x0080 0000	Table 262
PIO0_8	RW	0x020	PIO0_8/SPI MISO	0x0000 0000	Table 263
PIO0_9	RW	0x024	PIO0_9/SPI MOSI	0x0000 0000	Table 264
PIO0_10	RW	0x028	PIO0_10/CT32B_M0/SWCLK	0x0080 0002	Table 265
PIO0_11	RW	0x02C	PIO0_11/CT32B_M1/SWDIO	0x0080 0002	Table 266
ANA0_0	RW	0x030	ana_bus0	0x0000 0000	Table 267
ANA0_1	RW	0x034	ana_bus1	0x0000 0000	Table 268
ANA0_2	RW	0x038	ana_bus2	0x0000 0000	Table 269
ANA0_3	RW	0x03C	ana_bus3	0x0000 0000	Table 270
ANA0_4	RW	0x040	ana_bus4	0x0000 0000	Table 271
ANA0_5	RW	0x044	ana_bus5	0x0000 0000	Table 272
ANABUSGROUND	RW	0x080	analog bus grounding control	0x0FFF FFFF	Table 273

[1] If the WAKEUP bit is set in the PCON register of the PMU, the PIO0_0 WAKEUP function is automatically selected when entering deep power-down mode. The WAKEUP signal must be a stable low for 100 µs to wake up the system.

23.3.1 PIO0_0

Table 255. PIO0_0 register (address 0x4004 4000) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_0	
		001b	WAKEUP	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		00b	disable	
		01b	enable	
31:6	-	reserved		0

23.3.2 PIO0_1

Table 256. PIO0_1 register (address 0x4004 4004) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_1	
		001b	CLKOUT	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control)		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
31:6	-	reserved		0

23.3.3 PIO0_2

Table 257. PIO0_2 register (address 0x4004 4008) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_2	
		001b	SPI/SSP SSEL	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control)		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
31:6	-	reserved		0

23.3.4 PIO0_3

PIO0_3 is a high-drive, dual-mode (voltage and current sink) source/sink pad.

Table 258. PIO0_3 register (address 0x4004 400C) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_3	
		001b	CT16B_M0	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control)		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	

Table 258. PIO0_3 register (address 0x4004 400C) bit description*...continued*

Bit	Symbol	Description		Reset value
6	CDRIVE	Select pin output driver		0
	-	0	fixed-voltage driver	
		1	programmable current driver	
7	DDRIVE	Digital drive strength		0
		0	high drive	
		1	ultra-high drive	
15:8	ILO	Output LOW (deasserted) value		0
23:16	IHI	Output HIGH (asserted) value		80h
31:24	-	reserved		0

23.3.5 PIO0_4

Table 259. PIO0_4 register (address 0x4004 4010) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		1
		000b	PIO0_4 (open-drain pin)	
		001b	I^2C-bus SCL (open-drain pin)	
		(other)	(not allowed)	
7:3	-	reserved		0
9:8	I2CMODE	Selects the output mode of the pad.		0
		00b	I^2C-bus standard/fast-mode	
		01b	open-drain GPIO (4 mA) functionality	
		10b	reserved	
		11b	reserved	
31:10	-	reserved		0

23.3.6 PIO0_5

Table 260. PIO0_5 register (address 0x4004 4014) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		1
		000b	PIO0_5 (open-drain pin)	
		001b	I^2C SDA (open-drain pin)	
		(other)	(not allowed)	
7:3	-	reserved		0

Table 260. PIO0_5 register (address 0x4004 4014) bit description*...continued*

Bit	Symbol	Description		Reset value
9:8	I2CMODE	Selects the output mode of the pad.		0
		00b	I^2C-bus standard/fast-mode	
		01b	open-drain GPIO (4 mA) functionality	
		10b	reserved	
		11b	reserved	
31:10	-	reserved		0

23.3.7 PIO0_6

Table 261. PIO0_6 register (address 0x4004 4018) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_6	
		001b	SPI/SSP SCLK	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
31:6	-	reserved		0

23.3.8 PIO0_7

PIO0_7 is a high-drive, dual-mode (voltage and current sink) source/sink pad.

Table 262. PIO0_7 register (address 0x4004 401C) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_7	
		001b	CT16B_M1	
		(other)	(not allowed)	

Table 262. PIO0_7 register (address 0x4004 401C) bit description*...continued*

Bit	Symbol	Description		Reset value
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
6	CDRIVE	Select pin output driver.		0
		0	fixed voltage driver	
		1	programmable current driver	
7	DDRIVE	Digital drive strength		-
		0	high drive	
		1	ultra-high drive	
15:8	ILO	Output LOW (deasserted) value		0
23:16	IHI	Output HIGH (asserted) value		80h
31:24	-	reserved		0

23.3.9 PIO0_8

Table 263. PIO0_8 register (address 0x4004 4020) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_8	
		001b	SPI/SSP MISO	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
31:6	-	reserved		0

23.3.10 PIO0_9

Table 264. PIO0_9 register (address 0x4004 4024) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	PIO0_9	
		001b	SPI/SSP MOSI	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
31:6	-	reserved		0

23.3.11 PIO0_10

PIO0_10 is a high-drive, dual-mode (voltage and current sink) source/sink pad.

Table 265. PIO0_10 register (address 0x4004 4028) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		010b
		000b	PIO0_10	
		001b	CT32B_M0	
		010b	SWCLK	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
6	CDRIVE	Select pin output driver.		0
		0	fixed-voltage driver	
		1	programmable current driver	
7	DDRIVE	Digital drive strength		0
		0	high drive	
		1	ultra-high drive	
15:8	ILO	Output LOW (deasserted) value		0
23:16	IHI	Output HIGH (asserted) value		80h
31:24	-	reserved		0

23.3.12 PIO0_11

PIO0_11 is a high-drive, dual-mode (voltage and current sink) source/sink pad.

Table 266. PIO0_11 register (address 0x4004 402C) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		010b
		000b	PIO0_11	
		001b	CT32B_M1	
		010b	SWDIO	
		(other)	(not allowed)	
4:3	MODE	Selects function mode (on-chip pull-up/pull-down resistor control).		0
		00b	inactive (no pull-up/down resistor enabled)	
		01b	pull-down resistor enabled	
		10b	pull-up resistor enabled	
		11b	repeater mode	
5	LPF	Low-pass filter		0
		0	disable	
		1	enable	
6	CDRIVE	Select pin output driver.		0
		0	fixed-voltage driver	
		1	programmable current driver	
7	DDRIVE	digital drive strength		0
		0	high drive pad	
		1	ultra-high drive pad	
15:8	ILO	Output LOW (deasserted) value		0
23:16	IHI	Output HIGH (asserted) value		80h
31:24	-	reserved		0

23.3.13 ANA0_0

Table 267. ANA0_0 register (address 0x4004 4030) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	connect to VSS	
		001b	connect to ana_extbus0	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.14 ANA0_1

Table 268. ANA0_1 register (address 0x4004 4034) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	connect to VSS	
		001b	connect to ana_extbus1	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.15 ANA0_2

Table 269. ANA0_2 register (address 0x4004 4038) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	connect to VSS	
		001b	connect to ana_extbus2	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.16 ANA0_3

Table 270. ANA0_3 register (address 0x4004 403C) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	connect to VSS	
		001b	connect to ana_extbus3	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.17 ANA0_4

Table 271. ANA0_4 register (address 0x4004 4040) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved		0
		000b	connect to VSS	
		001b	connect to ana_extbus4	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.18 ANA0_5

Table 272. ANA0_5 register (address 0x4004 4044) bit description

Bit	Symbol	Description		Reset value
2:0	FUNC	Selects pin function. All other values are reserved.		0
		000b	connect to VSS	
		001b	connect to ana_extbus5	
		(other)	(not allowed)	
31:3	-	reserved		0

23.3.19 ANABUSGROUND

The ANABUSGROUND register controls the internal grounding of the analog buses. By default, all analog buses are connected to VSS.

Table 273. ANABUSGROUND register (address 0x4004 408) bit description

Bit	Symbol	Description		Reset value
0	AGND_E0	Connect ana_extbus0 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
1	AGND_E1	Connect ana_extbus1 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
2	AGND_E2	Connect ana_extbus2 to V.SS		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
3	AGND_E3	Connect ana_extbus3 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	

Table 273. ANABUSGROUND register (address 0x4004 408) bit description *...continued*

Bit	Symbol	Description		Reset value
4	AGND_E4	Connect ana_extbus4 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
5	AGND_E5	Connect ana_extbus5 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
6	AGND_E6	Connect ana_extbus6 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
7	AGND_E7	Connect ana_extbus7 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
8	AGND_E8	Connect ana_extbus8 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
9	AGND_E9	Connect ana_extbus9 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
10	AGND_E10	Connect ana_extbus10 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
11	AGND_E11	Connect ana_extbus11 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
12	AGND_I0	Connect ana_intbus0 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
13	AGND_I1	Connect ana_intbus1 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
14	AGND_I2	Connect ana_intbus2 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
15	AGND_I3	Connect ana_intbus3 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	

Table 273. ANABUSGROUND register (address 0x4004 408) bit description *...continued*

Bit	Symbol	Description		Reset value
16	AGND_I4	Connect ana_intbus4 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
17	AGND_5	Connect ana_intbus5 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
18	AGND_I6	Connect ana_intbus6 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
19	AGND_I7	Connect ana_intbus7 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
20	AGND_I8	Connect ana_intbus8 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
21	AGND_I9	Connect ana_intbus9 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
22	AGND_I10	Connect ana_intbus10 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
23	AGND_I11	Connect ana_intbus11 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
24	AGND_I12	Connect ana_intbus12 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
25	AGND_I13	Connect ana_intbus13 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
26	AGND_I14	Connect ana_intbus14 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	
27	AGND_I15	Connect ana_intbus15 to VSS.		1
		0	bus not connected to VSS	
		1	bus connected to VSS	

24 Nested vectored interrupt controller

24.1 About this chapter

This chapter describes the nested vectored interrupt controller (NVIC).

The NVIC is identical on members of the NHS31xx family.

24.2 Features

- Nested vectored interrupt controller that is a part of the Arm Cortex-M0+
- Tightly coupled interrupt controller provides low interrupt latency
- Controls system exceptions and peripheral interrupts
- The NVIC supports 32 vectored interrupts
- Four programmable interrupt priority levels with hardware priority level masking
- Software interrupt generation

24.3 General description

The NVIC is a part of the Arm Cortex-M0+. The processor automatically stacks its state on exception entry and unstacks it on exception exit, with no instruction overhead. This action provides low latency exception handling.

24.3.1 Interrupt sources

Table 274 lists the interrupt sources for each peripheral function. Each peripheral device may have one or more interrupt lines to the NVIC. Each line may represent more than one interrupt source. Interrupts with the same priority level are serviced in the order of their interrupt number.

See Section 24.4 for a detailed NVIC register description.

Table 274. Connection of interrupt sources to the NVIC

Exception number	Function	Flags
0-12	Start logic wake-up interrupts	Each interrupt connected to a PIO0 input pin serving as wake-up from deep-sleep mode. Interrupt 0 to 10 correspond to PIO0_0 to PIO0_10. Interrupt 11 corresponds to RFID/NFC external access. Interrupt 12 corresponds to the RTC timer.
13	RFID/NFC	RFID/NFC access detected/command received/read acknowledge.
14	RTC timer	RTC timer event interrupt
15	I^2C	SI (state change)
16	CT16B	Match 0 – 2
17	PMU	Power from field detected
18	CT32B	Match 0 – 1
19	BOD	Brownout detect (power drop)
20	SPI/SSP	TX FIFO half empty/RX FIFO half full/RX timeout/RX overrun
21	TSENS	Temperature sensor end of conversion/low threshold/high threshold
22	-	reserved

Table 274. Connection of interrupt sources to the NVIC...*continued*

Exception number	Function	Flags
23	-	reserved
24	I2D	Current-to-digital end of conversion/low threshold / high threshold
25	ADC/DAC	ADC/DAC interrupt
26	WDT	Watchdog interrupt (WDINT)
27	Flash	Flash memory
28	EEPROM	EEPROM memory
29-30	-	reserved
31	PIO0	GPIO interrupt status of port 0

24.4 Register descriptions

The NVIC registers are on the Arm private peripheral bus (APB).

Table 275. Register overview: NVIC (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value	Reference
ISER0	RW	0x100	Interrupt set enable register 0. This register allows enabling interrupts and reading back the interrupt enables for specific peripheral functions.	0	Table 276
-	-	0x104	reserved	-	-
ICER0	RW	0x180	Interrupt clear enable register 0. This register allows disabling interrupts and reading back the interrupt enables for specific peripheral functions.	0	Table 277
-	-	0x184	reserved	-	-
ISPR0	RW	0x200	Interrupt set pending register 0. This register allows changing the interrupt state to pending and reading back the interrupt pending state for specific peripheral functions.	0	Table 278
-	-	0x204	reserved	-	-
ICPR0	RW	0x280	Interrupt clear pending register 0. This register allows changing the interrupt state to not pending and reading back the interrupt pending state for specific peripheral functions.	0	Table 279
-	-	0x284	reserved	-	-
IABR0	RO	0x300	Interrupt active bit register 0. This register allows reading the current interrupt active state for specific peripheral functions.	0	Table 280
-	-	0x304	reserved	-	-
IPR0	RW	0x400	Interrupt priority registers 0. This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 0 to 3.	0	Table 281
IPR1	RW	0x404	Interrupt priority registers 1 This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 4 to 7.	0	Table 282
IPR2	RW	0x408	Interrupt priority registers 2. This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 8 to 11.	0	Table 283

Table 275. Register overview: NVIC (base address 0xE000 E000)...*continued*

Name	Access	Address offset	Description	Reset value	Reference
IPR3	RW	0x40C	Interrupt priority registers 3. This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 12 to 15.	0	Table 284
-	-	0x410	reserved	-	-
-	-	0x414	reserved	-	-
IPR6	RW	0x418	Interrupt priority registers 6. This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 24 to 27.	0	Table 285
IPR7	RW	0x41C	Interrupt priority registers 7. This register allows assigning a priority to each interrupt. It contains the 2-bit priority fields for interrupts 28 to 31.	0	Table 286

24.4.1 Interrupt set enable register 0 (ISER0)

The ISER0 register enables peripheral interrupts or contains the enabled state of those interrupts. Interrupts are disabled through the ICER0 register (see Table 277).

The bit description is as follows for all bits in this register:

- Write
  Writing logic 0 has no effect. Writing logic 1 disables the interrupt.
- Read
  Logic 0 indicates that the interrupt is disabled. Logic 1 indicates that the interrupt is enabled.

Table 276. ISER0 register (address 0xE000 E100) bit description

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt disable	0
1	ICE_SPI1	Interrupt disable	0
2	-	reserved	-
3	ICE_UART0	Interrupt disable	0
4	ICE_UART1	Interrupt disable	0
5	ICE_UART2	Interrupt disable	0
6	-	reserved	-
7	-	reserved	-
8	ICE_I2C	Interrupt disable	0
9	ICE_SCT	Interrupt disable	0
10	ICE_MRT	Interrupt disable	0
11	ICE_CMP	Interrupt disable	0
12	ICE_WDT	Interrupt disable	0
13	ICE_BOD	Interrupt disable	0
14	ICE_FLASH	Interrupt disable	0
15	ICE_WKT	Interrupt disable	0
16-23	-	reserved	-
24	ICE_PININT0	Interrupt disable	0

Table 276. ISER0 register (address 0xE000 E100) bit description...*continued*

Bit	Symbol	Description	Reset value
25	ICE_PININT1	Interrupt disable	0
26	ICE_PININT2	Interrupt disable	0
27	ICE_PININT3	Interrupt disable	0
28	ICE_PININT4	Interrupt disable	0
29	ICE_PININT5	Interrupt disable	0
30	ICE_PININT6	Interrupt disable	0
31	ICE_PININT7	Interrupt disable	0

24.4.2 Interrupt clear enable register 0 (ICER0)

The ICER0 register allows disabling the peripheral interrupts or for reading the enabled state of those interrupts. Enable interrupts through the ISER0 registers (see Table 276).

The bit description is as follows for all bits in this register:

- Write
  Writing logic 0 has no effect. Writing logic 1 disables the interrupt.
- Read
  Logic 0 indicates that the interrupt is disabled. Logic 1 indicates that the interrupt is enabled.

Table 277. ICER0 register (address 0xE000 E180) bit description

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt disable	0
1	ICE_SPI1	Interrupt disable	0
2	-	reserved	-
3	ICE_UART0	Interrupt disable	0
4	ICE_UART1	Interrupt disable	0
5	ICE_UART2	Interrupt disable	0
6	-	reserved	-
7	-	reserved	-
8	ICE_I2C	Interrupt disable	0
9	ICE_SCT	Interrupt disable	0
10	ICE_MRT	Interrupt disable	0
11	ICE_CMP	Interrupt disable	0
12	ICE_WDT	Interrupt disable	0
13	ICE_BOD	Interrupt disable	0
14	ICE_FLASH	Interrupt disable	0
15	ICE_WKT	Interrupt disable	0
16-23	-	reserved	-
24	ICE_PININT0	Interrupt disable	0

Table 277. ICER0 register (address 0xE000 E180) bit description...*continued*

Bit	Symbol	Description	Reset value
25	ICE_PININT1	Interrupt disable	0
26	ICE_PININT2	Interrupt disable	0
27	ICE_PININT3	Interrupt disable	0
28	ICE_PININT4	Interrupt disable	0
29	ICE_PININT5	Interrupt disable	0
30	ICE_PININT6	Interrupt disable	0
31	ICE_PININT7	Interrupt disable	0

24.4.3 Interrupt set pending register 0 (ISPR0)

The ISPR0 register allows setting the pending state of the peripheral interrupts or for reading the pending state of those interrupts. Clear the pending state of interrupts through the ICPR0 registers (see Table 279).

The bit description is as follows for all bits in this register:

- Write
  Writing logic 0 has no effect. Writing logic 1 changes the interrupt state to pending.
- Read
  Logic 0 indicates that the interrupt is not pending. Logic 1 indicates that the interrupt is pending.

Table 278. ISPR0 register (address 0xE000 E200) bit description

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt pending set	0
1	ICE_SPI1	Interrupt pending set	0
2	-	reserved	-
3	ICE_UART0	Interrupt pending set	0
4	ICE_UART1	Interrupt pending set	0
5	ICE_UART2	Interrupt pending set	0
6	-	reserved	-
7	-	reserved	-
8	ICE_I2C	Interrupt pending set	0
9	ICE_SCT	Interrupt pending set	0
10	ICE_MRT	Interrupt pending set	0
11	ICE_CMP	Interrupt pending set	0
12	ICE_WDT	Interrupt pending set	0
13	ICE_BOD	Interrupt pending set	0
14	ICE_FLASH	Interrupt pending set	0
15	ICE_WKT	Interrupt pending set	0
16-23	-	reserved	-
24	ICE_PININT0	Interrupt pending set	0

Table 278. ISPR0 register (address 0xE000 E200) bit description...*continued*

Bit	Symbol	Description	Reset value
25	ICE_PININT1	Interrupt pending set	0
26	ICE_PININT2	Interrupt pending set	0
27	ICE_PININT3	Interrupt pending set	0
28	ICE_PININT4	interrupt pending set	0
29	ICE_PININT5	Interrupt pending set	0
30	ICE_PININT6	Interrupt pending set	0
31	ICE_PININT7	Interrupt pending set	0

24.4.4 Interrupt clear pending register 0 (ICPR0)

The ICPR0 register allows clearing the pending state of the peripheral interrupts or for reading the pending state of those interrupts. Set the pending state of interrupts through the ISPR0 register (see Table 278).

The bit description is as follows for all bits in this register:

- Write
  Writing logic 0 has no effect. Writing logic 1 changes the interrupt state to not pending.
- Read
  Logic 0 indicates that the interrupt is not pending. Logic 1 indicates that the interrupt is pending.

Table 279. ICPR0 register (address 0xE000 E280) bit description

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt pending clear	0
1	ICE_SPI1	Interrupt pending clear	0
2	-	reserved	-
3	ICE_UART0	Interrupt pending clear	0
4	ICE_UART1	Interrupt pending clear	0
5	ICE_UART2	Interrupt pending clear	0
6	-	reserved	-
7	-	reserved	-
8	ICE_I2C	interrupt pending clear	0
9	ICE_SCT	Interrupt pending clear	0
10	ICE_MRT	Interrupt pending clear	0
11	ICE_CMP	Interrupt pending clear	0
12	ICE_WDT	interrupt pending clear	0
13	ICE_BOD	Interrupt pending clear	0
14	ICE_FLASH	interrupt pending clear	0
15	ICE_WKT	Interrupt pending clear	0
16-23	-	reserved	-
24	ICE_PININT0	interrupt pending clear	0

Table 279. ICPR0 register (address 0xE000 E280) bit description...*continued*

Bit	Symbol	Description	Reset value
25	ICE_PININT1	Interrupt pending clear	0
26	ICE_PININT2	Interrupt pending clear	0
27	ICE_PININT3	Interrupt pending clear	0
28	ICE_PININT4	Interrupt pending clear	0
29	ICE_PININT5	Interrupt pending clear	0
30	ICE_PININT6	Interrupt pending clear	0
31	ICE_PININT7	Interrupt pending clear	0

24.4.5 Interrupt active bit register 0 (IABR0)

The IABR0 register is a read-only register that allows reading the active state of the peripheral interrupts. To determine which peripherals assert an interrupt to the NVIC and may also be pending if they are enabled, use this register.

The bit description is as follows for all bits in this register:

- Write
  n/a.
- Read
  Logic 0 indicates that the interrupt is not active. Logic 1 indicates that the interrupt is active.

Table 280. IABR0 register (address 0xE000 E300) bit description

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt active	0
1	ICE_SPI1	Interrupt active	0
2	-	reserved	-
3	ICE_UART0	Interrupt active	0
4	ICE_UART1	Interrupt active	0
5	ICE_UART2	Interrupt active	0
6	-	reserved	-
7	-	reserved	-
8	ICE_I2C	Interrupt active	0
9	ICE_SCT	Interrupt active	0
10	ICE_MRT	Interrupt active	0
11	ICE_CMP	interrupt active	0
12	ICE_WDT	Interrupt active	0
13	ICE_BOD	Interrupt active	0
14	ICE_FLASH	Interrupt active	0
15	ICE_WKT	Interrupt active	0
16-23	-	reserved	-
24	ICE_PININT0	Interrupt active	0

Table 280. IABR0 register (address 0xE000 E300) bit description...*continued*

Bit	Symbol	Description	Reset value
25	ICE_PININT1	Interrupt active	0
26	ICE_PININT2	Interrupt active	0
27	ICE_PININT3	Interrupt active	0
28	ICE_PININT4	Interrupt active	0
29	ICE_PININT5	Interrupt active	0
30	ICE_PININT6	interrupt active	0
31	ICE_PININT7	Interrupt active	0

24.4.6 Interrupt priority register 0 (IPR0)

The IPR0 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 281. IPR0 register (address 0xE000 E400) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.
6:7	IP_SPI0	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.
14:15	IP_SPI1	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	-	reserved
24:29	-	These bits ignore writes and read as logic 0.
30:31	IP_UART0	Interrupt priority 0 = highest priority 3 = lowest priority

24.4.7 Interrupt priority register 1 (IPR1)

The IPR1 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 282. IPR1 register (address 0xE000 E404) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.
6:7	IP_UART1	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.

Table 282. IPR1 register (address 0xE000 E404) bit description*...continued*

Bit	Symbol	Description
14:15	IP_UART2	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	-	reserved
24:29	-	These bits ignore writes and read as logic 0.
30:31	-	reserved

24.4.8 Interrupt priority register 2 (IPR2)

The IPR2 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 283. IPR2 register (address 0xE000 E408) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.
6:7	IP_I2C	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.
14:15	IP_SCT	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	IP_MRT	Interrupt priority 0 = highest priority 3 = lowest priority
24:29	-	These bits ignore writes and read as logic 0.
30:31	IP_CMP	Interrupt priority 0 = highest priority 3 = lowest priority

24.4.9 Interrupt priority register 3 (IPR3)

The IPR3 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 284. IPR3 register (address 0xE000 E40C) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.
6:7	IP_WDT	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.

Table 284. IPR3 register (address 0xE000 E40C) bit description...*continued*

Bit	Symbol	Description
14:15	IP_BOD	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	IP_FLASH	Interrupt priority 0 = highest priority 3 = lowest priority
24:29	-	These bits ignore writes and read as logic 0.
30:31	IP_WKT	Interrupt priority 0 = highest priority 3 = lowest priority

24.4.10 Interrupt priority register 6 (IPR6)

The IPR6 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 285. IPR6 register (address 0xE000 E418) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.
6:7	IP_PININT0	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.
14:15	IP_PININT1	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	IP_PININT2	Interrupt priority 0 = highest priority 3 = lowest priority
24:29	-	These bits ignore writes and read as logic 0.
30:31	IP_PININT3	Interrupt priority 0 = highest priority 3 = lowest priority

24.4.11 Interrupt priority register 7 (IPR7)

The IPR7 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 286. IPR7 register (address 0xE000 E41C) bit description

Bit	Symbol	Description
0:5	-	These bits ignore writes and read as logic 0.

Table 286. IPR7 register (address 0xE000 E41C) bit description*...continued*

Bit	Symbol	Description
6:7	IP_PININT4	Interrupt priority 0 = highest priority 3 = lowest priority
8:13	-	These bits ignore writes and read as logic 0.
14:15	IP_PININT5	Interrupt priority 0 = highest priority 3 = lowest priority
16:21	-	These bits ignore writes and read as logic 0.
22:23	IP_PININT6	Interrupt priority 0 = highest priority 3 = lowest priority
24:29	-	These bits ignore writes and read as logic 0.
30:31	IP_PININT7	Interrupt priority 0 = highest priority 3 = lowest priority

25 Soldering guidelines

25.1 HVQFN24

The packaging and soldering information is available in SOT616-3.

Note: *For the HVQFN24 package, any introduction of mechanical stress may affect the temperature accuracy negatively.*

- Use a standard PCB reflow oven process. Manual soldering introduces mechanical stress.
- When used on a rigid PCB:
  To minimize mechanical stress, do not solder the exposed pad.
- When used on a flexible foil:
  To increase board level reliability, solder the exposed pad. If soldered, it must be connected to ground or left floating.

25.2 WLCSP25

The packaging information is available in SOT1401-1.

The soldering information is available in the "Wafer-level chip-scale package (fan-in WLP and fan-out WLP)" application note (Ref. 7). The NHS31xx WLCSP25 package uses bump pitches of 400 µm.

26 Abbreviations

Table 287. Abbreviations

Acronym	Description
ADC	analog-to-digital converter
AHB	advanced high-performance bus
AMBA	advanced microcontroller bus architecture
APB	advanced peripheral bus
API	application programming interface
Arm	acorn RISC machine
CGU	clock generator unit
DAC	digital-to-analog converter
EEPROM	electrically erasable programmable read-only memory
FIFO	first in first out
JTAG	joint test action group
LDO	low drop out regulator
NDEF	NFC data exchange format
NFC	near field communication
NVMC	non-volatile memory controller
PMU	power management unit
RISC	reduced instruction set computer
RTC	real-time clock
SPI	serial peripheral interface
SWD	serial wire debug
SSI	synchronous serial interface

27 References

[1] **AN11340 application note** — MIFARE Ultralight and MIFARE Ultralight EV1 features and hints; 2021, NXP Semiconductors

[2] **MF0ULx1** — MIFARE Ultralight EV1 - Contactless ticket IC; 2019, NXP Semiconductors

[3] **DUI 0662B (ID011713)** — Cortex-M0+ Devices - Generic User Guide; 2012, Arm

[4] **DDI 0484C (ID011713)** — Cortex-M0+ Devices - Technical Reference Manual; 2012, Arm

[5] **AN11657 application note** — NHS31xx temperature sensor calibration; 2021, NXP Semiconductors

[6] **ISO/IEEE 11073-10472** — Health informatics - Point-of-care medical device communication - Device specialization - Medication monitor; 2018, ISO

[7] **AN10439 application note** — Wafer-level chip-scale package (fan-in WLP and fan-out WLP); 2018, NXP Semiconductors

28 Legal information

28.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

28.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

28.3 Licenses

Purchase of NXP ICs with NFC technology

Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

28.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

Tables

Tab. 1. Revision history 2
Tab. 2. Feature overview 5
Tab. 3. Ordering Information 5
Tab. 4. Pad allocation table of the HVQFN24 package 10
Tab. 5. Pad description of the HVQFN24 package 11
Tab. 6. Ball allocation table of the WLCSP25 package 13
Tab. 7. Ball description of the WLCSP25 package 13
Tab. 8. Bump allocation table of the NHS3100W8 package 15
Tab. 9. Bump description of the NHS3100W8 package 15
Tab. 10. Register overview system configuration control block - SYSCON (base address 0x4004 8000) 19
Tab. 11. SYSMEMREMAP register (address 0x4004 8000) bit description 20
Tab. 12. PRESETCTRL register (address 0x4004 8004) bit description 20
Tab. 13. SYSCLKCTRL register (address 0x4004 8020) bit description 21
Tab. 14. Impact of different clock frequency settings 22
Tab. 15. SYSCLKUEN register (address 0x4004 8024) bit description 22
Tab. 16. SYSRSTSTAT register (address 0x4004 8030) bit description 23
Tab. 17. SYSAHBCLKCTRL register (address 0x4004 8080) bit description 24
Tab. 18. SSPCLKDIV register (address 0x4004 8094) bit description 25
Tab. 19. WDTCLKSEL register (address 0x4004 80D0) bit description 26
Tab. 20. WDTCLKUEN register (address 0x4004 80D4) bit description 26
Tab. 21. WDTCLKDIV register (address 0x4004 80D8) bit description 26
Tab. 22. CLKOUTEN register (address 0x4004 80E8) bit description 27
Tab. 23. SYSTCKCAL register (address 0x4004 8154) bit description 27
Tab. 24. STARTAPRP0 register (address 0x4004 8200) bit description 27
Tab. 25. STARTERP0 register (address 0x4004 8204) bit description 28
Tab. 26. STARTRSRP0CLR register (address 0x4004 8208) bit description 28
Tab. 27. STARTSRP0 register (address 0x4004 820C) bit description 29
Tab. 28. PDAWAKECFG register (address 0x4004 8234) bit description 29
Tab. 29. PDRUNCFG register (address 0x4004 8238) bit description 30
Tab. 30. DEVICE_ID register (address 0x4004 83F4) bit description 31
Tab. 31. Device IDs 31
Tab. 32. IC power states 38
Tab. 33. State transition events for DEEPSLEEP to ACTIVE 38
Tab. 34. State transition events for DEEP POWER-DOWN to ACTIVE 39
Tab. 35. Register overview: PMU (base address 0x4003 8000) 40
Tab. 36. PCON register (address 0x4003 8000) bit description 40
Tab. 37. GPREG0 - GPREG4 registers (address 0x4003 8004 to 0x4003 8014) bit description 42
Tab. 38. PSTAT register (address 0x4003 8018) bit description 42
Tab. 39. ACCSTAT register (address 0x4003 8020) bit description 43
Tab. 40. TMRCLKCTRL register (address 0x4003 802C) bit description 44
Tab. 41. IMSC register (address 0x4003 8030) bit description 44
Tab. 42. RIS register (address 0x4003 8034) bit description 44
Tab. 43. MIS register (address 0x4003 8038) bit description 44
Tab. 44. ICR register (address 0x4003 803C) bit description 45
Tab. 45. IAP command list 47
Tab. 46. IAP Read factory settings command 48
Tab. 47. Supported registers for factory setting read 49
Tab. 48. IAP Prepare sectors for write operation command 49
Tab. 49. IAP Copy data to flash command 49
Tab. 50. IAP Erase sector command 50
Tab. 51. IAP Blank-check sector command 50
Tab. 52. IAP Read part identification command 51
Tab. 53. IAP Read boot code version number command 51
Tab. 54. IAP Compare command 51
Tab. 55. IAP Read Unique Identification number command 52
Tab. 56. IAP Erase page command 52
Tab. 57. IAP Status Codes 53
Tab. 58. Conversion time for different resolution of TSENS 54
Tab. 59. Register overview for temperature sensor (base address 0x4006 0000) 55
Tab. 60. CR register (address 0x4006 0000) bit description 55
Tab. 61. DR register (address 0x4006 0004) bit description 55
Tab. 62. SR register (address 0x4006 0008) bit description 56
Tab. 63. SP0 register (address 0x4006 000C) bit description 56
Tab. 64. SP1 register (address 0x4006 0010) bit description 57

Tab. 65. SP2 register (address 0x4006 0014) bit description 57
Tab. 66. SP3 register (address 0x4006 0018) bit description 57
Tab. 67. TLO register (address 0x4006 001C) bit description 57
Tab. 68. THI register (address 0x4006 0020) bit description 58
Tab. 69. IMSC register (address 0x4006 0024) bit description 58
Tab. 70. RIS register (address 0x4006 0028) bit description 58
Tab. 71. MIS register (address 0x4006 002C) bit description 59
Tab. 72. ICR register (address 0x4006 0030) bit description 59
Tab. 73. Connections to the Current-to-Digital analog input multiplexer 60
Tab. 74. ENOB for different input ranges and corrections 60
Tab. 75. Register overview: Current-to-Digital converter (base address 0x4006 8000) 64
Tab. 76. CR register (address 0x4006 8000) bit description 64
Tab. 77. DR register (address 0x4006 8004) bit description 65
Tab. 78. SR register (address 0x4006 8008) bit description 65
Tab. 79. SP0 register (address 0x4006 800C) bit description 65
Tab. 80. SP1 register (address 0x4006 8010) bit description 66
Tab. 81. SP2 register (address 0x4006 8014) bit description 66
Tab. 82. SP3 register (address 0x4006 8018) bit description 67
Tab. 83. MUX register (address 0x4006 801C) bit description 67
Tab. 84. TLO register (address 0x4006 4020) bit description 67
Tab. 85. THI register (address 0x4006 4024) bit description 68
Tab. 86. IMSC register (address 0x4006 8028) bit description 68
Tab. 87. RIS register (address 0x4006 802C) bit description 68
Tab. 88. MIS register (address 0x4006 8030) bit description 69
Tab. 89. ICR register (address 0x4006 4034) bit description 69
Tab. 90. Connections to ADC input and output multiplexers 70
Tab. 91. Register overview: Analog-to-Digital / Digital-to-Analog converter (base address 0x4001 C000) 71
Tab. 92. CR register (address 0x4001 C000) bit description 72
Tab. 93. ADCDR register (address 0x4001 C004) bit description 72
Tab. 94. DACDR register (address 0x4001 C008) bit description 72
Tab. 95. SR register (address 0x4001 C00C) bit description 72
Tab. 96. ADCSP register (address 0x4001 C010) bit description 73
Tab. 97. ADCMUX register (address 0x4001 C018) bit description 73
Tab. 98. DACMUX register (address 0x4001 C01C) bit description 74
Tab. 99. ADCTLO register (address 0x4001 C020) bit description 74
Tab. 100. ADC0THI register (address 0x4001 C024) bit description 74
Tab. 101. IMSC register (address 0x4001 C028) bit description 74
Tab. 102. RIS register (address 0x4001 C02C) bit description 75
Tab. 103. MIS register (address 0x4001 C030) bit description 75
Tab. 104. ADC0ICR register (address 0x4001 C034) bit description 76
Tab. 105. GET_VERSION response for NHS devices 80
Tab. 106. RFID/NFC communication block register and buffer memory space overview: APB side 80
Tab. 107. RFID/NFC communication block special registers and buffer memory space overview: RFID side 81
Tab. 108. CFG register (address 0x4005 8000) bit description 81
Tab. 109. SR register (APB address 0x4005 8004, RF address 0x84) bit description 82
Tab. 110. CMDIN register (APB address 0x4005 800C, RF address 0x85) bit description 83
Tab. 111. CMDOUT register (APB address 0x4005 8010, RF address 0x86) bit description 83
Tab. 112. TARGET register (address 0x4005 8010) bit description 83
Tab. 113. LAST_ACCESS register (address 0x4005 8014) bit description 83
Tab. 114. IMSC register (address 0x4005 8014) bit description 83
Tab. 115. RIS register (address 0x4005 801C) bit description 84
Tab. 116. ICR register (address 0x4005 8024) bit description 84
Tab. 117. Register overview: RTC timer (base address 0x4005 4000) 86
Tab. 118. RTC timer control register (address 0x4005 4000) bit description 87
Tab. 119. RTC Timer status register (address 0x4005 4004) bit description 87
Tab. 120. RTC timer calibration register (address 0x4005 4008) bit description 88
Tab. 121. RTC timer sleep time register (address 0x4005 400C) bit description 88
Tab. 122. RTC timer current value register (address 0x4005 4010) bit description 88

Tab. 123. IMSC register (address 0x4005 4014) bit description 88
Tab. 124. RIS register (address 0x4005 4018) bit description 89
Tab. 125. MIS register (address 0x4005 401C) bit description 89
Tab. 126. ICR register (address 0x4005 4020) bit description 89
Tab. 127. ACCSTAT register (address 0x4005 4024) bit description 90
Tab. 128. TIME register (address 0x4005 4030) bit description 91
Tab. 129. Register overview: Watchdog timer block (base address 0x4000 4000) 93
Tab. 130. WDMOD register (address 0x4000 4000) bit description 94
Tab. 131. Watchdog operating mode selection 94
Tab. 132. WDTC register (address 0x4000 4004) bit description 95
Tab. 133. WDFEED register (address 0x4000 4008) bit description 95
Tab. 134. WDTV register (address 0x4000 400C) bit description 96
Tab. 135. Register overview: SYSTICK timer (base address 0xE000 E000) 99
Tab. 136. CSR register (address 0xE000 E010) bit description 100
Tab. 137. RVR register (address 0xE000 E014) bit description 100
Tab. 138. CVR register (address 0xE000 E018) bit description 100
Tab. 139. CALIB register (address 0xE000 E01C) bit description 101
Tab. 140. Counter/timer pin description 103
Tab. 141. Register overview: 16-bit timer 0 CT16B0 (base address 0x4000 C000) 106
Tab. 142. IR register (address 0x4000 C000) bit description 107
Tab. 143. TCR register (address 0x4000 C004) bit description 107
Tab. 144. TC register (address 0x4000 C008) bit description 107
Tab. 145. PR register (address 0x4000 C00C) bit description 108
Tab. 146. PC register (address 0x4001 C010) bit description 108
Tab. 147. MCR register (address 0x4000 C014) bit description 108
Tab. 148. MR0/1/2/3 registers (addresses 0x4000 C018 to 24) bit description 109
Tab. 149. EEMR register (address 0x4000 C03C) bit description 110
Tab. 150. PWMC register (address 0x4000 C074) bit description 111
Tab. 151. Counter/timer pin description 114
Tab. 152. Register overview: 32-bit counter/timer 0 CT32B (base address 0x4000 4000) 116
Tab. 153. IR register (address 0x4000 4000) bit description 117
Tab. 154. TCR register (address 0x4000 4004) bit description 117
Tab. 155. TC register (address 0x4000 4008) bit description 117
Tab. 156. PR register (address 0x4000 400C) bit description 118
Tab. 157. PC register (address 0x4001 4010) bit description 118
Tab. 158. MCR register (address 0x4000 4014) bit description 118
Tab. 159. MR0/1/2/3 registers (addresses 0x4000 4018 to 24) bit description 120
Tab. 160. EMR register (address 0x4000 403C) bit description 120
Tab. 161. PWMC register (address 0x4000 4074) bit description 121
Tab. 162. Register overview: EEPROM memory controller (base address 0x4003 4000) 124
Tab. 163. RWSTATE register (address 0x4003 4008) bit description 125
Tab. 164. WSTATE register (address 0x4003 4010) bit description 125
Tab. 165. PAUTOPROG register (address 0x4003 400C) bit description 126
Tab. 166. CLKDIV register (address 0x4003 4014) bit description 126
Tab. 167. CLKDIV1 register (address 0x4003 4054) bit description 126
Tab. 168. CLKDIV2 register (address 0x4003 4058) bit description 127
Tab. 169. PWRDWN register (address 0x4003 4020) bit description 127
Tab. 170. MSSTART register (address 0x4003 4020) bit description 127
Tab. 171. MSSTOP register (address 0x4003 4024) bit description 127
Tab. 172. MSDATASIG register (address 0x4003 4028) bit description 128
Tab. 173. MSPARSIG register (address 0x4003 402C) bit description 128
Tab. 174. STATUS register (address 0x4003 4034) bit description 128
Tab. 175. INT_CLR_ENABLE register (address 0x4003 4FD8) bit description 129
Tab. 176. INT_SET_ENABLE register (address 0x4003 4FDC) bit description 129
Tab. 177. INT_CLR_STATUS register (address 0x4003 4FE8) bit description 129
Tab. 178. INT_ENABLE register (address 0x4003 4FE0) bit description 129
Tab. 179. INT_STATUS register (address 0x4003 4FE4) bit description 129
Tab. 180. MODULE_ID register (address 0x4003 4FFC) bit description 130
Tab. 181. Feedback bits for MISR computation 135
Tab. 182. Register overview: Flash memory controller (base address 0x4004 C000) 135
Tab. 183. FCTR register (address 0x4003 C000) bit description 136

Tab. 184. FSTAT register (address 0x4003 C010) bit description 137
Tab. 185. FBWST register (address 0x4003 C010) bit description 137
Tab. 186. WST values for different Flash operating modes (FCTR settings) and system clock speed 138
Tab. 187. FPTR register (address 0x4003 C008) bit description 138
Tab. 188. FCRA register (address 0x4003 C01C) bit description 138
Tab. 189. FMSSTART register (address 0x4003 C020) bit description 139
Tab. 190. FMSSTOP register (address 0x4003 C024) bit description 139
Tab. 191. FMS16 register (address 0x4003 C028) bit description 139
Tab. 192. FMSW0 register (address 0x4003 C02C) bit description 139
Tab. 193. ECCRSTERRCNT register (address 0x4003 C050) bit description 139
Tab. 194. ECCERRCNT register (address 0x4003 C054) bit description 140
Tab. 195. INT_CLR_ENABLE register (address 0x4003 CFD8) bit description 140
Tab. 196. INT_SET_ENABLE register (address 0x4003 CFDC) bit description 140
Tab. 197. INT_CLR_STATUS register (address 0x4003 CFE8) bit description 140
Tab. 198. INT_SET_STATUS register (address 0x4003 CFEC) bit description 140
Tab. 199. INT_ENABLE register (address 0x4003 CFE0) bit description 141
Tab. 200. INT_STATUS register (address 0x4003 CFE4) bit description 141
Tab. 201. MODULE_ID register (address 0x4003 CFFC) bit description 141
Tab. 202. I2C-bus pin description 143
Tab. 203. Register overview of the I2C-bus controller (base 0x4000 000) 144
Tab. 204. CONSET register (address 0x4000 0000) bit description 145
Tab. 205. STAT register (address 0x4000 0004) bit description 147
Tab. 206. DAT register (address 0x4000 0008) bit description 147
Tab. 207. ADR0 register (address 0x4000 000C) bit description 147
Tab. 208. SCLH register (address 0x4000 0010) bit description 148
Tab. 209. SCLL register (address 0x4000 0014) bit description 148
Tab. 210. SCLL + SCLH values for selected I2C-bus clock rates 148
Tab. 211. CONCLR register (address 0x4000 0018) bit description 148
Tab. 212. MMCTRL register (address 0x4000 001C) bit description 149
Tab. 213. ADR1/2/3 registers (addresses 0x4000 0020,0024,0028) bit description 151
Tab. 214. DATA_BUFFER register (address 0x4000 002C) bit description 151
Tab. 215. MASK0/1/2/3 register (addresses 0x4000 0030,0034,0038,003C) bit description 151
Tab. 216. CONSET used to initialize master transmitter mode 152
Tab. 217. Master transmitter mode 153
Tab. 218. CONSET used to configure slave mode 155
Tab. 219. Abbreviations used to describe an I2C-bus operation 161
Tab. 220. CONSET used to initialize master transmitter mode 161
Tab. 221. Master transmitter mode 162
Tab. 222. Master receive mode 165
Tab. 223. ADR usage in slave receiver mode 166
Tab. 224. CONSET used to initialize slave transmitter mode 167
Tab. 225. Slave receiver mode 167
Tab. 226. Slave transmitter mode 171
Tab. 227. Miscellaneous states 173
Tab. 228. SPI pin description 179
Tab. 229. Register overview: Serial peripheral interface (base address 0x4004 0000) 187
Tab. 230. CR0 register (address 0x4004 0000) bit description 188
Tab. 231. CR1 register (address 0x4004 0004) bit description 189
Tab. 232. DR register 0 (address 0x4004 0008) bit description 189
Tab. 233. SR register (address 0x4004 000C) bit description 190
Tab. 234. CPSR register (address 0x4004 0010) bit description 190
Tab. 235. IMSC register (address 0x4004 0014) bit description 190
Tab. 236. RIS register (address 0x4004 0018) bit description 191
Tab. 237. MIS register (address 0x4004 001C) bit description 191
Tab. 238. ICR register (address 0x4004 0020) bit description 192
Tab. 239. Register overview: General-purpose I/O (base address 0x5000 0000) 194
Tab. 240. GPIODATA register (address 0x5000 0000 to 0x5000 3FFC) bit description 195
Tab. 241. DIR register (address 0x5000 8000) bit description 195
Tab. 242. IS register (address 0x5000 8004) bit description 195
Tab. 243. IBE register (address 0x5000 8008) bit description 196
Tab. 244. IEV register (address 0x5000 800C) bit description 196
Tab. 245. IMSC register (address 0x5000 8010) bit description 196
Tab. 246. RIS register (address 0x5000 8014) bit description 196

Tab. 247. MIS register (address 0x5000 8018) bit description 197
Tab. 248. IC register (address 0x5000 801C) bit description 197
Tab. 249. Pad direction and MFIO settings 198
Tab. 250. LPF bit and MFIO pad output selection 198
Tab. 251. Pad output behavior 199
Tab. 252. DDRIVE bit settings 200
Tab. 253. Pad mode as function of CDRIVE 200
Tab. 254. Register overview: I/O configuration (base address 0x4004 4000) 201
Tab. 255. PIO0_0 register (address 0x4004 4000) bit description 202
Tab. 256. PIO0_1 register (address 0x4004 4004) bit description 202
Tab. 257. PIO0_2 register (address 0x4004 4008) bit description 203
Tab. 258. PIO0_3 register (address 0x4004 400C) bit description 203
Tab. 259. PIO0_4 register (address 0x4004 4010) bit description 204
Tab. 260. PIO0_5 register (address 0x4004 4014) bit description 204
Tab. 261. PIO0_6 register (address 0x4004 4018) bit description 205
Tab. 262. PIO0_7 register (address 0x4004 401C) bit description 205
Tab. 263. PIO0_8 register (address 0x4004 4020) bit description 206
Tab. 264. PIO0_9 register (address 0x4004 4024) bit description 207
Tab. 265. PIO0_10 register (address 0x4004 4028) bit description 208
Tab. 266. PIO0_11 register (address 0x4004 402C) bit description 209
Tab. 267. ANA0_0 register (address 0x4004 4030) bit description 209
Tab. 268. ANA0_1 register (address 0x4004 4034) bit description 210
Tab. 269. ANA0_2 register (address 0x4004 4038) bit description 210
Tab. 270. ANA0_3 register (address 0x4004 403C) bit description 210
Tab. 271. ANA0_4 register (address 0x4004 4040) bit description 211
Tab. 272. ANA0_5 register (address 0x4004 4044) bit description 211
Tab. 273. ANABUSGROUND register (address 0x4004 408) bit description 211
Tab. 274. Connection of interrupt sources to the NVIC 214
Tab. 275. Register overview: NVIC (base address 0xE000 E000) 215
Tab. 276. ISER0 register (address 0xE000 E100) bit description 216
Tab. 277. ICER0 register (address 0xE000 E180) bit description 217
Tab. 278. ISPR0 register (address 0xE000 E200) bit description 218
Tab. 279. ICPR0 register (address 0xE000 E280) bit description 219
Tab. 280. IABR0 register (address 0xE000 E300) bit description 220
Tab. 281. IPR0 register (address 0xE000 E400) bit description 221
Tab. 282. IPR1 register (address 0xE000 E404) bit description 221
Tab. 283. IPR2 register (address 0xE000 E408) bit description 222
Tab. 284. IPR3 register (address 0xE000 E40C) bit description 222
Tab. 285. IPR6 register (address 0xE000 E418) bit description 223
Tab. 286. IPR7 register (address 0xE000 E41C) bit description 223
Tab. 287. Abbreviations 226

Figures

Fig. 1. NHS3100 block diagram 5
Fig. 2. NHS3152 block diagram 6
Fig. 3. NHS31xx family memory map 7
Fig. 4. Pin configuration of HVQFN24 package 10
Fig. 5. Ball configuration of WLCSP25 package 12
Fig. 6. Bump configuration Bump die: Top view, bumps up 15
Fig. 7. NHS31xx clock generator block diagram 18
Fig. 8. NHS31xx power architecture 37
Fig. 9. Power rails and power supply pad types 37
Fig. 10. PMU state transition diagram 38
Fig. 11. NHS31xx power-up sequence 39
Fig. 12. IAP command passing 46
Fig. 13. Current-to-Digital block diagram 60
Fig. 14. Typical offset deviation due to temperature changes, relative to the offset at 25 °C and the selected range 61
Fig. 15. Typical gain deviation due to temperature changes, relative to the gain at 25 °C and the selected range 62
Fig. 16. Typical offset deviation due to temperature changes, relative to the offset at 25 °C and the selected range 62
Fig. 17. Typical gain deviation due to temperature changes, relative to the gain at 25 °C and the selected range 63
Fig. 18. Analog-to-Digital / Digital-to-Analog block diagram 70
Fig. 19. NHS31xx analog bus structure 77
Fig. 20. Block diagram of the RFID/NFC interface 78

Fig. 21. Memory map as seen from RF interface 79
Fig. 22. System tick timer block diagram 98
Fig. 23. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled 103
Fig. 24. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled 104
Fig. 25. 16-bit timer block diagram 105
Fig. 26. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR2) and MAT2:0 enabled as PWM outputs by the PWCM register 112
Fig. 27. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled 114
Fig. 28. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled 115
Fig. 29. 32-bit counter/timer block diagram 115
Fig. 30. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR2) and MAT2:0 enabled as PWM outputs by the PWCM register 122
Fig. 31. Typical I2C-bus configuration 142
Fig. 32. Format and states in the master transmitter mode 154
Fig. 33. Data format of master receiver mode 155
Fig. 34. Master receiver switches to master transmitter after sending Repeated-START ...155
Fig. 35. Data format of slave receiver mode 156
Fig. 36. Data format of slave mode 156
Fig. 37. I2C serial interface block diagram 157
Fig. 38. Arbitration procedure 159
Fig. 39. Serial clock synchronization 159
Fig. 40. Format and states in the master transmitter mode 164
Fig. 41. Format and states in Master receiver mode .. 166
Fig. 42. Format and states in slave receiver mode 170
Fig. 43. Format and states in slave transmitter mode 172
Fig. 44. Simultaneous Repeated-START conditions from two masters 174
Fig. 45. Forced access to a busy I2C-bus 175
Fig. 46. Recovering from a bus obstruction caused by a LOW level on SDA 175
Fig. 47. Texas Instruments synchronous Serial Frame Format: Single frame transfer (top), back-to-back two frames transfer (bottom) 181
Fig. 48. SPI bus timing 182
Fig. 49. SPI frame format with CPOL = logic 0 and CPHA = logic 0. (a) Single and b) Continuous Transfer 182
Fig. 50. SPI frame format with CPOL = logic 0 and CPHA = logic 1 183
Fig. 51. SPI frame format with CPOL = logic 1 and CPHA = logic 0 (a) Single and b) Continuous Transfer) 184
Fig. 52. SPI frame format with CPOL = logic 1 and CPHA = logic 1 185
Fig. 53. Microwire frame format (single transfer) 186
Fig. 54. Microwire frame format (continuous transfers) 186
Fig. 55. Masked write operation to the GPIODATA register 193
Fig. 56. Masked GPIO read operation 194
Fig. 57. Pin configuration with current drive mode 200

Contents

1 Introductory information 3
1.1 Introduction 3
1.2 Attributes 3
1.3 Device-dependent features 5
1.4 Ordering information 5
1.5 Block diagram 5
1.6 Arm Cortex-M0+ core configuration 6
1.7 Memory map 7
1.7.1 Memories 8
1.7.1.1 SRAM (0x1000 0000 to 0x1000 1FFF) 8
1.7.1.2 Flash (0x0000 0000 to 0x0000 7FFF) 8
1.7.1.3 EEPROM (0x3000 0000 to 0x3000 0FFF) 8
2 Product feature overview 9
2.1 About this section 9
2.2 General description 9
2.2.1 NHS3100 for cold chain monitoring and therapy adherence 9
2.2.2 NHS3152 for therapy adherence using resistive sensing 9
3 Pinning 10
3.1 About this section 10
3.2 HVQFN24 10
3.3 WLCSP25 12
3.4 NHS3100W8 15
4 System configuration (SYSCON) 17
4.1 General description 17
4.2 Clock generation 17
4.3 System FRO clock specifications 18
4.3.1 SFRO trimming 18
4.4 Timer FRO clock specifications 18
4.5 Register descriptions 18
4.5.1 System memory remap register (SYSMEMREMAP) 20
4.5.2 Peripheral reset control register (PRESETCTRL) 20
4.5.3 System clock control register (SYSCLKCTRL) 21
4.5.4 System clock update enable register (SYSCLKUEN) 22
4.5.5 System reset status register (SYSRSTSTAT) 23
4.5.6 AHB clock control register 24
4.5.7 SPI/SSP clock divider control register (SSPCLKDIV) 25
4.5.8 Watchdog timer clock source selection register (WDTCLKSEL) 25
4.5.9 Watchdog timer clock update enable register (WDTCLKUEN) 26
4.5.10 Watchdog timer clock divider register (WDTCLKDIV) 26
4.5.11 Clock output enable register (CLKOUTEN) 27
4.5.12 System tick counter calibration register (SYSTCKCAL) 27
4.5.13 Start logic edge control register 0 (STARTAPRP0) 27
4.5.14 Start logic signal enable register 0 (STARTERP0) 28
4.5.15 Start logic reset register 0 (STARTRSRP0CLR) 28
4.5.16 Start logic status register 0 (STARTSRP0) 29
4.5.17 Wake-up configuration register (PDAWAKECFG) 29
4.5.18 Power-down configuration register (PDRUNCFG) 30
4.5.19 Device ID register (DEVICE_ID) 31
5 Power management unit (PMU) 32
5.1 About this section 32
5.2 General description 32
5.3 Functional description 32
5.3.1 Active mode 32
5.3.1.1 Power configuration in Active mode 33
5.3.2 Sleep mode 33
5.3.3 Deep-sleep mode 33
5.3.4 Deep power-down mode 34
5.4 System power architecture 35
5.5 Brownout detection 39
5.6 Register descriptions 40
5.6.1 Power control register (PCON) 40
5.6.2 General-purpose registers 0 to 4 (GPREG0-4) 42
5.6.3 Power status register (PSTAT) 42
5.6.4 Access status register (ACCSTAT) 43
5.6.5 Timer clock control register (TMRCLKCTRL) 43
5.6.6 PMU interrupt mask set and clear register (IMSC) 44
5.6.7 PMU raw interrupt status register (RIS) 44
5.6.8 PMU masked interrupt status register (MIS) 44
5.6.9 PMU interrupt clear register (ICR) 45
6 In-application programming (IAP) firmware 46
6.1 IAP command list 47
6.1.1 Read factory settings 48
6.1.2 Prepare sectors for write operation 49
6.1.3 Copy data to flash 49
6.1.4 Erase sectors 50
6.1.5 Blank-checking sectors 50
6.1.6 Read part identification number 51
6.1.7 Read boot code version number 51
6.1.8 Compare 51
6.1.9 Read UID 52
6.1.10 Erase page 52
6.2 IAP status codes 53
7 Temperature sensor 54
7.1 About this section 54
7.2 Introduction 54
7.3 Features 54
7.4 General description 54
7.5 Calibration 54
7.6 Register descriptions 55
7.6.1 Temperature sensor control register (CR) 55
7.6.2 Temperature sensor data register (DR) 55

7.6.3 Temperature sensor status register (SR) 56
7.6.4 Temperature sensor setup register 0 (SP0) 56
7.6.5 Temperature sensor setup register 1 (SP1) 57
7.6.6 Temperature sensor setup register 2 (SP2) 57
7.6.7 Temperature sensor setup register 3 (SP2) 57
7.6.8 Temperature sensor low-temperature threshold register (TLO) 57
7.6.9 Temperature sensor high-temperature threshold register (THI) 58
7.6.10 Temperature sensor interrupt mask set/clear register (IMSC) 58
7.6.11 Temperature sensor raw interrupt status register (RIS) 58
7.6.12 Temperature sensor register masked interrupt status register (MIS) 59
7.6.13 Temperature sensor register interrupt clear register (ICR) 59
8 Current-to-digital converter (I2D) 60
8.1 About this chapter 60
8.2 Introduction 60
8.3 Input multiplexer 60
8.4 ENOB 60
8.5 Impact of temperature on offset and gain error 61
8.6 Register descriptions 64
8.6.1 Current-to-digital control register (CR) 64
8.6.2 Current-to-digital data register (DR) 65
8.6.3 Current-to-digital status register (SR) 65
8.6.4 Current-to-digital setup register 0 (SP0) 65
8.6.5 Current-to-digital setup register 1 (SP1) 66
8.6.6 Current-to-digital setup register 2 (SP2) 66
8.6.7 Current-to-digital setup register 3 (SP3) 67
8.6.8 Current-to-digital input multiplexer (MUX) 67
8.6.9 Current-to-digital low-value threshold register (TLO) 67
8.6.10 Current-to-digital high-value threshold register (THI) 68
8.6.11 Current-to-digital interrupt mask Set/Clear register (IMSC) 68
8.6.12 Current-to-digital raw interrupt status register (RIS) 68
8.6.13 Current-to-digital masked interrupt status register (MIS) 69
8.6.14 Current-to-digital interrupt clear register (ICR) 69
9 Analog-to-digital/digital-to-analog converter (ADCDAC0) 70
9.1 About this chapter 70
9.2 Features 70
9.3 General description 70
9.4 Input multiplexer and output switch matrix 70
9.5 Register descriptions 71
9.5.1 ADC control register (CR) 72
9.5.2 ADC data register (ADCDR) 72
9.5.3 DAC data register (DACDR) 72
9.5.4 ADC status register (SR) 72
9.5.5 ADCDAC setup register (ADCSP) 73
9.5.6 ADC input multiplexer register (ADCMUX) 73
9.5.7 DAC output switch register (DACMUX) 74
9.5.8 ADCDAC low-value threshold register (ADCTLO) 74
9.5.9 ADCDAC high-value threshold register (ADCTHI) 74
9.5.10 ADCDAC interrupt mask set/clear register (IMSC) 74
9.5.11 ADCDAC raw interrupt status register (RIS) 75
9.5.12 ADCDAC masked interrupt status register (MIS) 75
9.5.13 ADCDAC interrupt clear register (ICR) 76
10 Analog signal buses 77
10.1 About this chapter 77
10.2 General description 77
11 RFID/NFC communication unit 78
11.1 About this chapter 78
11.2 Introduction 78
11.3 Features 78
11.4 General description 78
11.5 Pin description 79
11.6 Functional description 79
11.6.1 ISO 14443 A and MIFARE operation 79
11.6.2 NHS31xx family communication protocol 80
11.6.2.1 Device-specific GET_VERSION command 80
11.7 Register descriptions 80
11.7.1 RFID/NFC configuration register (CFG) 81
11.7.2 RFID/NFC status register (SR) 82
11.7.3 RFID/NFC incoming command register (CMDIN) 82
11.7.4 RFID/NFC outgoing data register (CMDOUT) 83
11.7.5 RFID/NFC target address register (TARGET) 83
11.7.6 RFID/NFC last accessed address register (LAST_ACCESS) 83
11.7.7 RFID/NFC interrupt mask register (IMSC) 83
11.7.8 RFID/NFC raw interrupt status register (RIS) 84
11.7.9 RFID/NFC interrupt clear register (ICR) 84
11.7.10 RFID/NFC buffer memory 85
12 Real-time clock (RTC) 86
12.1 About this chapter 86
12.2 Features 86
12.3 General description 86
12.4 Register descriptions 86
12.4.1 RTC control register (CR) 87
12.4.2 RTC status register (SR) 87
12.4.3 RTC calibration register (CAL) 87
12.4.4 RTC timer sleep time register (SLEEPT) 88
12.4.5 RTC timer current value register (VAL) 88
12.4.6 RTC timer interrupt mask set/clear register (IE) 88
12.4.7 RTC timer raw interrupt status register (RIS) 88
12.4.8 RTC timer masked interrupt status register (MIS) 89
12.4.9 RTC timer interrupt clear register (ICR) 89
12.4.10 RTC timer access status register (ACCSTAT) 90
12.4.11 RTC timer time register (TIME) 91

13 WatchDog Timer (WDT) 92
13.1 About this section 92
13.2 Purpose 92
13.2.1 Configuration 92
13.2.2 Features 92
13.2.3 General description 92
13.3 Register descriptions 93
13.3.1 Watchdog mode register 94
13.3.2 Watchdog timer constant register 95
13.3.3 Watchdog feed register 95
13.3.4 Watchdog timer value register 96
14 SYSTICK timer 97
14.1 About this section 97
14.2 Purpose 97
14.3 Features 97
14.4 General description 98
14.5 Functional description 99
14.5.1 Example timer calculation 99
14.6 Register descriptions 99
14.6.1 System timer control and status register (CSR) 100
14.6.2 System timer reload value register (RVR) 100
14.6.3 System timer current value register (CVR) 100
14.6.4 System timer calibration value register (CALIB) 101
15 16-bit timer CT16B 102
15.1 About this section 102
15.2 Configuration 102
15.3 Features 102
15.4 Applications 102
15.5 General description 103
15.6 Pin description 103
15.7 Example timer operation 103
15.8 Architecture 105
15.9 Register descriptions 106
15.9.1 Interrupt register (IR) 107
15.9.2 Timer control register (TCR) 107
15.9.3 Timer counter register (TC) 107
15.9.4 Prescale register (PR) 108
15.9.5 Prescale counter register (PC) 108
15.9.6 Match control register (MCR) 108
15.9.7 Match register (MR0/1/2/3) 109
15.9.8 External match register (EMR) 110
15.9.9 PWM control register (PWMC) 111
15.9.10 Rules for single-edge controlled PWM outputs 112
16 32-bit timer CT32B 113
16.1 About this section 113
16.2 Configuration 113
16.3 Features 113
16.4 Applications 113
16.5 General description 114
16.6 Pin description 114
16.7 Example timer operation 114
16.8 Architecture 115
16.9 Register descriptions 116
16.9.1 Interrupt register (IR) 116
16.9.2 Timer control register (TCR) 117
16.9.3 Timer counter register (TC) 117
16.9.4 Prescale register (PR) 118
16.9.5 Prescale counter register (PC) 118
16.9.6 Match control register (MCR) 118
16.9.7 Match registers (MR0/1/2/3) 120
16.9.8 External match register (EMR) 120
16.9.9 PWM control register (PWMC) 121
16.9.10 Rules for single edge controlled PWM outputs 122
17 EEPROM controller 123
17.1 About this section 123
17.2 On-chip EEPROM 123
17.3 Functional description 123
17.3.1 Reading from EEPROM 123
17.3.2 Writing to EEPROM 124
17.4 Register descriptions 124
17.4.1 Wait state registers (WSTATE, WSTATE) 125
17.4.2 EEPROM auto-programming register (PAUTOPROG) 126
17.4.3 EEPROM program clock management registers (CLKDIV, CLKDIV1, CLKDIV2) 126
17.4.4 EEPROM power down register (PWRDWN) 127
17.4.5 Signature generation registers 127
17.4.6 Registers with status information 128
17.4.7 Interrupt registers 129
17.4.8 Module ID register (MODULE_ID) 130
18 Flash controller 131
18.1 About this chapter 131
18.2 On-chip flash 131
18.3 Features 131
18.4 Functional description 131
18.4.1 Reading from flash 131
18.4.2 Writing to flash 132
18.4.3 Erasing/programming flash 132
18.4.4 Signature computation 134
18.5 Register descriptions 135
18.5.1 Flash control register (FCTR) 136
18.5.2 Flash status register (FSTAT) 137
18.5.3 Flash wait state register (FBWST) 137
18.5.4 Registers for program/erase management 138
18.5.5 Signature generation registers 139
18.5.6 Registers with ECC status information 139
18.5.7 Interrupt registers 140
18.5.8 Module ID register (MODULE_ID) 141
19 I2C-bus interface 142
19.1 About this chapter 142
19.2 General description 142
19.3 Features 143
19.4 Configuration 143
19.5 Applications 143
19.6 Pin description 143
19.7 Register descriptions 144
19.7.1 I2C control set register (CONSET) 145
19.7.2 I2C status register (STAT) 147
19.7.3 I2C-bus data register (DAT) 147
19.7.4 I2C-bus slave address register 0 (ADR0) 147
19.7.5 I2C-bus SCL high and low duty cycle registers (SCLH / SCLL) 148
19.7.6 I2C-bus control clear register (CONCLR) 148

19.7.7 I2C-bus monitor mode control register (MMCTRL) ... 149
19.7.8 I2C-bus slave address registers 1:3 (ADR1/2/3) ... 150
19.7.9 I2C-bus data buffer register (DATA_ BUFFER) ... 151
19.7.10 I2C-bus mask registers 0:3 (MASK0/1/2/3) ... 151
19.8 I2C-bus operating modes ... 152
19.8.1 Master transmitter mode ... 152
19.8.2 Master receiver mode ... 155
19.8.3 Slave receiver mode ... 155
19.8.4 Slave transmitter mode ... 156
19.9 I2C-bus implementation and operation ... 157
19.9.1 Input filters and output stages ... 158
19.9.2 Address registers, ADDR0 to ADDR3 ... 158
19.9.3 Address mask registers, MASK0 to MASK3 .. 158
19.9.4 Comparator ... 158
19.9.5 Shift register, DAT ... 158
19.9.6 Arbitration and synchronization logic ... 158
19.9.7 Serial clock generator ... 159
19.9.8 Timing and control ... 160
19.9.9 Control register; CONSET and CONCLR ... 160
19.9.10 Status decoder and status register ... 160
19.10 Details of I2C-bus operating modes ... 161
19.10.1 Master transmitter mode ... 161
19.10.2 Master receiver mode ... 164
19.10.3 Slave receiver mode ... 166
19.10.4 Slave transmitter mode ... 170
19.10.5 Miscellaneous states ... 173
19.10.5.1 STAT = 0xF8 ... 173
19.10.5.2 STAT = 0x00 ... 173
19.10.6 Some special cases ... 173
19.10.6.1 Simultaneous Repeated-START conditions from two masters ... 173
19.10.6.2 Data transfer after loss of arbitration ... 174
19.10.6.3 Forced access to the I2C-bus ... 174
19.10.6.4 I2C-bus obstructed by a low level on SCL or SDA ... 175
19.10.6.5 Bus error ... 176
19.10.7 I2C-bus state service routines ... 176
19.10.8 Initialization ... 176
19.10.9 I2C-bus interrupt service ... 176
19.10.10 The state service routines ... 176
19.10.11 Adapting state services to an application ... 177
20 Serial wire debug (SWD) ... 178
20.1 About this chapter ... 178
20.2 Introduction ... 178
20.3 Features ... 178
21 Serial peripheral interface (SPI) ... 179
21.1 About this chapter ... 179
21.2 Features ... 179
21.3 General description ... 179
21.4 Configuration ... 179
21.5 Pin description ... 179
21.5.1 Pin detailed description ... 180
21.6 Functional description ... 180
21.6.1 Texas Instruments synchronous serial frame format ... 180
21.6.2 SPI frame format ... 181
21.6.2.1 Clock Polarity (CPOL) and phase (CPHA) control ... 181
21.6.2.2 SPI format with CPOL = logic 0, CPHA = logic 0 ... 182
21.6.2.3 SPI format with CPOL = logic 0, CPHA = logic 1 ... 183
21.6.2.4 SPI format with CPOL = logic 1, CPHA = logic 0 ... 184
21.6.2.5 SPI format with CPOL = logic 1, CPHA = logic 1 ... 185
21.6.3 Semiconductor Microwire frame format ... 186
21.7 Register descriptions ... 187
21.7.1 SPI/SSP control register 0 (CR0) ... 188
21.7.2 SPI/SSP control register 1 (CR1) ... 189
21.7.3 SPI/SSP data register (DR) ... 189
21.7.4 SPI/SSP status register (SR) ... 190
21.7.5 SPI/SSP clock prescale register (CPSR) ... 190
21.7.6 SPI/SSP interrupt mask set/clear register (IMSC) ... 190
21.7.7 SPI/SSP raw interrupt status register (RIS) ... 191
21.7.8 SPI/SSP masked interrupt status register (MIS) ... 191
21.7.9 SPI/SSP interrupt clear register (ICR) ... 192
22 General-purpose input output (GPIO) ... 193
22.1 About this chapter ... 193
22.2 Features ... 193
22.3 General description ... 193
22.4 Functional description ... 193
22.4.1 Write/read data operation ... 193
22.4.1.1 Write operation ... 193
22.4.1.2 Read operation ... 194
22.5 Register descriptions ... 194
22.5.1 GPIO data register (GPIODATA) ... 194
22.5.2 GPIO data direction register (DIR) ... 195
22.5.3 GPIO interrupt sense register (IS) ... 195
22.5.4 GPIO interrupt both edges sense register (IBE) ... 196
22.5.5 GPIO interrupt event register (IEV) ... 196
22.5.6 GPIO interrupt mask register (IMSC) ... 196
22.5.7 GPIO raw interrupt status register (RIS) ... 196
22.5.8 GPIO masked interrupt status register (MIS) ... 197
22.5.9 GPIO interrupt clear register (IC) ... 197
23 Input/output configuration ... 198
23.1 About this chapter ... 198
23.2 General description ... 198
23.2.1 PIO0 pin functions ... 198
23.2.2 PIO0 pin mode ... 198
23.2.3 PIO0 I2C mode ... 199
23.2.4 PIO0 current drive mode ... 199
23.2.5 Reset state of pins ... 200
23.2.6 ANA0 input selection ... 201
23.3 IOCON register descriptions (base address 0x4004 4000) ... 201
23.3.1 PIO0_0 ... 202
23.3.2 PIO0_1 ... 202
23.3.3 PIO0_2 ... 203
23.3.4 PIO0_3 ... 203
23.3.5 PIO0_4 ... 204

23.3.6 PIO0_5 204
23.3.7 PIO0_6 205
23.3.8 PIO0_7 205
23.3.9 PIO0_8 206
23.3.10 PIO0_9 207
23.3.11 PIO0_10 208
23.3.12 PIO0_11 209
23.3.13 ANA0_0 209
23.3.14 ANA0_1 210
23.3.15 ANA0_2 210
23.3.16 ANA0_3 210
23.3.17 ANA0_4 211
23.3.18 ANA0_5 211
23.3.19 ANABUSGROUND 211
24 Nested vectored interrupt controller 214
24.1 About this chapter 214
24.2 Features 214
24.3 General description 214
24.3.1 Interrupt sources 214
24.4 Register descriptions 215
24.4.1 Interrupt set enable register 0 (ISER0) 216
24.4.2 Interrupt clear enable register 0 (ICER0) 217
24.4.3 Interrupt set pending register 0 (ISPR0) 218
24.4.4 Interrupt clear pending register 0 (ICPR0) 219
24.4.5 Interrupt active bit register 0 (IABR0) 220
24.4.6 Interrupt priority register 0 (IPR0) 221
24.4.7 Interrupt priority register 1 (IPR1) 221
24.4.8 Interrupt priority register 2 (IPR2) 222
24.4.9 Interrupt priority register 3 (IPR3) 222
24.4.10 Interrupt priority register 6 (IPR6) 223
24.4.11 Interrupt priority register 7 (IPR7) 223
25 Soldering guidelines 225
25.1 HVQFN24 225
25.2 WLCSP25 225
26 Abbreviations 226
27 References 227
28 Legal information 228

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 13 September 2021
Document identifier: UM10876

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP:

NHS3152THADADK NHS3152THADADKUL