nRF52840

Product Specification

v1.0

NORDIC® SEMICONDUCTOR

Feature list

Features:

- *Bluetooth*® 5, IEEE 802.15.4-2006, 2.4 GHz transceiver
  - -95 dBm sensitivity in 1 Mbps *Bluetooth*® low energy (BLE) mode
  - -103 dBm sensitivity in 125 kbps BLE mode (long range)
  - +8 dBm TX power (down to -20 dBm in 4 dB steps)
  - On-air compatible with nRF52, nRF51, nRF24L and nRF24AP Series
  - Supported data rates:
    - *Bluetooth*® 5: 2 Mbps, 1 Mbps, 500 kbps and 125 kbps
    - IEEE 802.15.4-2006: 250 kbps
    - Proprietary 2.4 GHz: 2 Mbps, 1 Mbps
  - Single-ended antenna output (on-chip balun)
  - 128-bit AES/ECB/CCM/AAR co-processor (on-the-fly packet encryption)
  - 4.8 mA peak current in TX (0 dBm)
  - 4.6 mA peak current in RX
  - RSSI (1 dB resolution)
- ARM® Cortex®-M4 32-bit processor with FPU, 64 MHz
  - 212 EEMBC CoreMark score running from flash memory
  - 52 µA/MHz running from flash memory
  - Watchpoint and trace debug modules (DWT, ETM and ITM)
  - Serial wire debug (SWD)
- Rich set of security features
  - ARM® TrustZone® Cryptocell 310 security subsystem
    - NIST SP800-90A and SP800-90B compliant random number generator
    - AES-128: ECB,CBC,CMAC/CBC-MAC,CTR,CCM/CCM*
    - Chacha20/Poly1305 AEAD supporting 128- and 256-bit key size
    - SHA-1, SHA-2 up to 256 bits
    - Keyed-hash message authentication code (HMAC)
    - RSA up to 2048-bit key size
    - SRP up to 3072-bit key size
    - ECC support for most used curves, among others P-256 (secp256r1) and Ed25519/Curve25519
    - Application key management using derived key model
  - Secure boot ready
    - Flash access control list (ACL)
    - Root-of-trust (RoT)
    - Debug control and configuration
    - Access port protection (CTRL-AP)
  - Secure erase
- Flexible power management
  - 1.7 V-5.5 V supply voltage range
  - On-chip DC/DC and LDO regulators with automated low current modes
  - 1.8 V-3.3 V regulated supply for external components
  - Automated peripheral power management
  - Fast wake-up using 64 MHz internal oscillator
  - 0.4 µA at 3 V in System OFF mode, no RAM retention
  - 1.5 µA at 3 V in System ON mode, no RAM retention, wake on RTC
- 1 MB flash and 256 kB RAM
- Advanced on-chip interfaces
  - USB 2.0 full speed (12 Mbps) controller
  - QSPI 32 MHz interface
  - High-speed 32 MHz SPI
  - Type 2 near field communication (NFC-A) tag with wake-on field
    - Touch-to-pair support
  - Programmable peripheral interconnect (PPI)
  - 48 general purpose I/O pins
  - EasyDMA automated data transfer between memory and peripherals
- Nordic SoftDevice ready with support for concurrent multi-protocol
- 12-bit, 200 ksps ADC - 8 configurable channels with programmable gain
- 64 level comparator
- 15 level low-power comparator with wake-up from System OFF mode
- Temperature sensor
- 4x 4-channel pulse width modulator (PWM) unit with EasyDMA
- Audio peripherals: I2S, digital microphone interface (PDM)
- 5x 32-bit timer with counter mode
- Up to 4x SPI master/3x SPI slave with EasyDMA
- Up to 2x I2C compatible 2-wire master/slave
- 2x UART (CTS/RTS) with EasyDMA
- Quadrature decoder (QDEC)
- 3x real-time counter (RTC)
- Single crystal operation
- Package variants
  - aQFN™ 73 package, 7 x 7 mm

Applications:

- Advanced computer peripherals and I/O devices
  - Mouse
  - Keyboard
  - Multi-touch trackpad
- Advanced wearables
  - Health/fitness sensor and monitor devices
  - Wireless payment enabled devices
- Internet of things (IoT)
  - Smart home sensors and controllers
  - Industrial IoT sensors and controllers
- Interactive entertainment devices
  - Remote controls
  - Gaming controllers

Contents

1 Revision history

Date	Version	Description
March 2018	1.0	First release

2 About this document

This product specification is organized into chapters based on the modules and peripherals that are available in this IC.

The peripheral descriptions are divided into separate sections that include the following information:

- A detailed functional description of the peripheral
- Register configuration for the peripheral
- Electrical specification tables, containing performance data which apply for the operating conditions described in Recommended operating conditions on page 544.

2.1 Document naming and status

Nordic uses three distinct names for this document, which are reflecting the maturity and the status of the document and its content.

Document name	Description
Objective Product Specification (OPS)	Applies to document versions up to 0.7. This product specification contains target specifications for product development.
Preliminary Product Specification (PPS)	Applies to document versions 0.7 and up to 1.0. This product specification contains preliminary data. Supplementary data may be published from Nordic Semiconductor ASA later.
Product Specification (PS)	Applies to document versions 1.0 and higher. This product specification contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Table 1: Defined document names

2.2 Peripheral naming and abbreviations

Every peripheral has a unique capitalized name or an abbreviation of its name, e.g. TIMER, used for identification and reference. This name is used in chapter headings and references, and it will appear in the ARM® Cortex® Microcontroller Software Interface Standard (CMSIS) hardware abstraction layer to identify the peripheral.

The peripheral instance name, which is different from the peripheral name, is constructed using the peripheral name followed by a numbered postfix, starting with 0, for example, TIMER0. A postfix is normally only used if a peripheral can be instantiated more than once. The peripheral instance name is also used in the CMSIS to identify the peripheral instance.

2.3 Register tables

Individual registers are described using register tables. These tables are built up of two sections. The first three colored rows describe the position and size of the different fields in the register. The following rows describe the fields in more detail.

2.3.1 Fields and values

The **Id (Field Id)** row specifies the bits that belong to the different fields in the register. If a field has enumerated values, then every value will be identified with a unique value id in the **Value Id** column.

A blank space means that the field is reserved and read as undefined, and it also must be written as `0` to secure forward compatibility. If a register is divided into more than one field, a unique field name is specified for each field in the **Field** column. The **Value Id** may be omitted in the single-bit bit fields when values can be substituted with a Boolean type enumerator range, e.g. true/false, disable(d)/enable(d), on/off, and so on.

Values are usually provided as decimal or hexadecimal. Hexadecimal values have a `0x` prefix, decimal values have no prefix.

The **Value** column can be populated in the following ways:

- Individual enumerated values, for example 1, 3, 9.
- Range of values, e.g. [0..4], indicating all values from and including 0 and 4.
- Implicit values. If no values are indicated in the **Value** column, all bit combinations are supported, or alternatively the field's translation and limitations are described in the text instead.

If two or more fields are closely related, the **Value Id**, **Value**, and **Description** may be omitted for all but the first field. Subsequent fields will indicate inheritance with '..'.

A feature marked **Deprecated** should not be used for new designs.

2.4 Registers

Register	Offset	Description
DUMMY	0x514	Example of a register controlling a dummy feature

Table 2: Register overview

2.4.1 DUMMY

Address offset: 0x514

Example of a register controlling a dummy feature

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID								D	D	D	D						C	C	C								B							A	A
Reset 0x00050002				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
ID	RW	Field	Value ID	Value				Description																											
A	RW	FIELD_A						Example of a field with several enumerated values																													
			Disabled	0				The example feature is disabled																													
			NormalMode	1				The example feature is enabled in normal mode																													
			ExtendedMode	2				The example feature is enabled along with extra functionality																													

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID								D	D	D	D					C	C	C									B							A	A
Reset 0x00050002				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
ID	RW	Field	Value ID	Value				Description																											
B	RW	FIELD_B						Example of a deprecated field																										Deprecated	
			Disabled	0				The override feature is disabled																											
			Enabled	1				The override feature is enabled																											
C	RW	FIELD_C						Example of a field with a valid range of values																											
			ValidRange	[2..7]				Example of allowed values for this field																											
D	RW	FIELD_D						Example of a field with no restriction on the values																											

3 Block diagram

This block diagram illustrates the overall system. Arrows with white heads indicate signals that share physical pins with other signals.

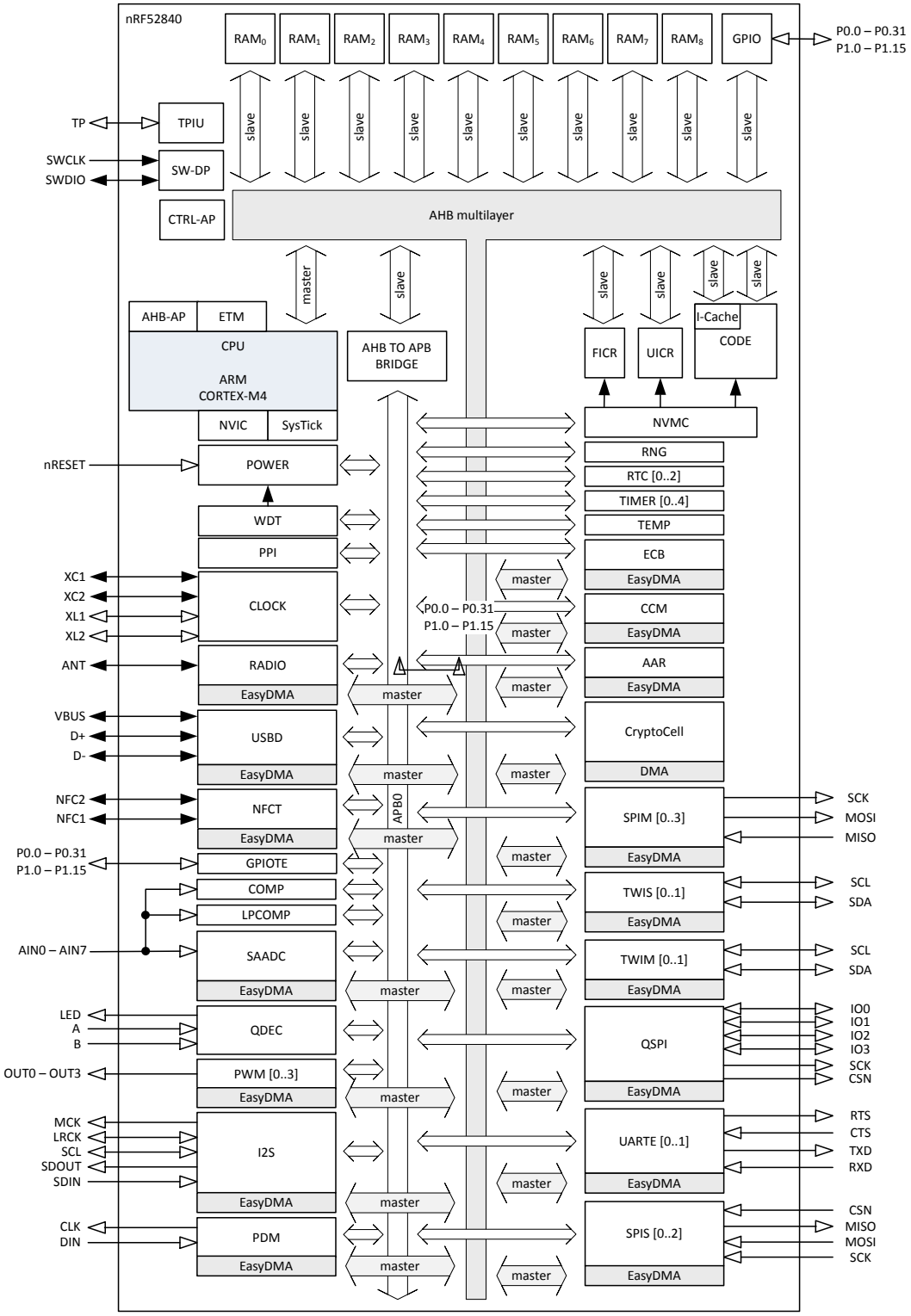

Figure 1: Block diagram

4 Core components

4.1 CPU

The ARM® Cortex®-M4 processor with floating-point unit (FPU) has a 32-bit instruction set (Thumb®-2 technology) that implements a superset of 16 and 32-bit instructions to maximize code density and performance.

This processor implements several features that enable energy-efficient arithmetic and high-performance signal processing, including:

- Digital signal processing (DSP) instructions
- Single-cycle multiply and accumulate (MAC) instructions
- Hardware divide
- 8- and 16-bit single instruction multiple data (SIMD) instructions
- Single-precision floating-point unit (FPU)

The ARM® Cortex® Microcontroller Software Interface Standard (CMSIS) hardware abstraction layer for the ARM®Cortex® processor series is implemented and available for the M4 CPU.

Real-time execution is highly deterministic in thread mode, to and from sleep modes, and when handling events at configurable priority levels via the nested vectored interrupt controller (NVIC).

Executing code from flash will have a wait state penalty on the nRF52 series. An instruction cache can be enabled to minimize flash wait states when fetching instructions. For more information on cache, see Cache on page 25. The section Electrical specification on page 19 shows CPU performance parameters including wait states in different modes, CPU current and efficiency, and processing power and efficiency based on the CoreMark® benchmark.

The ARM system timer (SysTick) is present on nRF52840. The SysTick's clock will only tick when the CPU is running or when the system is in debug interface mode.

4.1.1 Floating point interrupt

The floating point unit (FPU) may generate exceptions when used due to e.g. overflow or underflow, which in turn will trigger the FPU interrupt.

See Instantiation on page 22 for more information about the exceptions triggering the FPU interrupt.

To clear the IRQ (interrupt request) line when an exception has occurred, the relevant exception bit within the floating-point status and control register (FPSCR) needs to be cleared. For more information about the FPSCR or other FPU registers, see *Cortex-M4 Devices Generic User Guide*.

4.1.2 CPU and support module configuration

The ARM® Cortex®-M4 processor has a number of CPU options and support modules implemented on the device.

Option / Module	Description	Implemented
Core options		
NVIC	Nested vector interrupt controller	48 vectors
PRIORITIES	Priority bits	3
WIC	Wakeup interrupt controller	NO
Endianness	Memory system endianness	Little endian
Bit-banding	Bit banded memory	NO
DWT	Data watchpoint and trace	YES
SysTick	System tick timer	YES
Modules		
MPU	Memory protection unit	YES
FPU	Floating-point unit	YES
DAP	Debug access port	YES
ETM	Embedded trace macrocell	YES
ITM	Instrumentation trace macrocell	YES
TPIU	Trace port interface unit	YES
ETB	Embedded trace buffer	NO
FPB	Flash patch and breakpoint unit	YES
HTM	AMBA™ AHB trace macrocell	NO

4.1.3 Electrical specification

4.1.3.1 CPU performance

The CPU clock speed is 64 MHz. Current and efficiency data is taken when in System ON and the CPU is executing the CoreMark™ benchmark. It includes power regulator and clock base currents. All other blocks are IDLE.

Symbol	Description	Min.	Typ.	Max.	Units
W_{FLASH}	CPU wait states, running CoreMark from flash, cache disabled			2	
$W_{FLASHCACHE}$	CPU wait states, running CoreMark from flash, cache enabled			3	
W_{RAM}	CPU wait states, running CoreMark from RAM			0	
CM_{FLASH}	CoreMark, running CoreMark from flash, cache enabled		212		CoreM
$CM_{FLASH/MHz}$	CoreMark per MHz, running CoreMark from flash, cache enabled		3.3		CoreMark/MHz
$CM_{FLASH/mA}$	CoreMark per mA, running CoreMark from flash, cache enabled, DCDC 3V		59		CoreM mA

4.2 Memory

The nRF52840 contains 1 MB of flash and 256 kB of RAM that can be used for code and data storage.

The CPU and the peripherals having EasyDMA can access memory via the AHB multilayer interconnect.

The CPU is also able to access peripherals via the AHB multilayer interconnect, as illustrated in Memory layout on page 20.

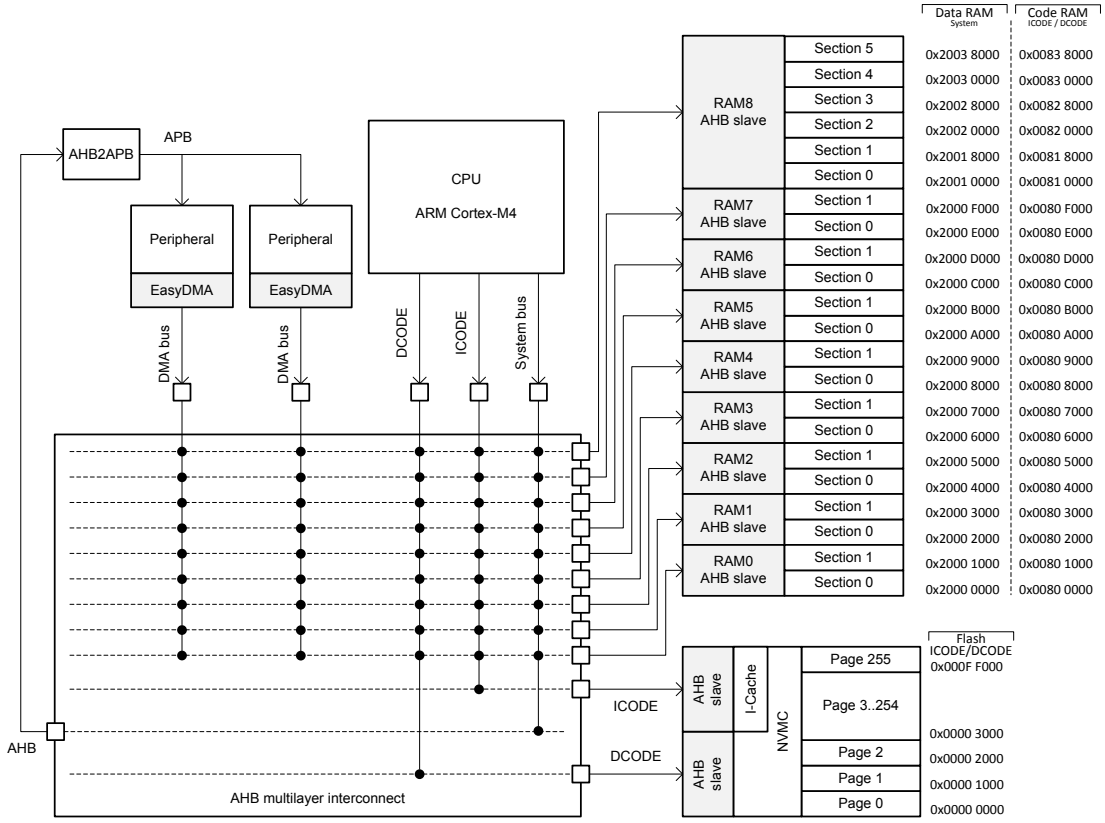

Figure 2: Memory layout

See AHB multilayer on page 48 and EasyDMA on page 45 for more information about the AHB multilayer interconnect and the EasyDMA.

The same physical RAM is mapped to both the Data RAM region and the Code RAM region. It is up to the application to partition the RAM within these regions so that one does not corrupt the other.

4.2.1 RAM - Random access memory

The RAM interface is divided into 9 RAM AHB slaves.

RAM AHB slave 0-7 is connected to 2x4 kB RAM sections each and RAM AHB slave 8 is connected to 6x32 kB sections, as shown in Memory layout on page 20.

Each of the RAM sections have separate power control for System ON and System OFF mode operation, which is configured via RAM register (see the POWER — Power supply on page 60).

4.2.2 Flash - Non-volatile memory

The flash can be read an unlimited number of times by the CPU, but it has restrictions on the number of times it can be written and erased and also on how it can be written.

Writing to flash is managed by the non-volatile memory controller (NVMC), see NVMC — Non-volatile memory controller on page 23.

The flash is divided into 256 pages of 4 kB each that can be accessed by the CPU via both the ICODE and DCODE buses as shown in Memory layout on page 20.

4.2.3 Memory map

The complete memory map is shown in Memory map on page 21. As described in Memory on page 19, Code RAM and the Data RAM are the same physical RAM.

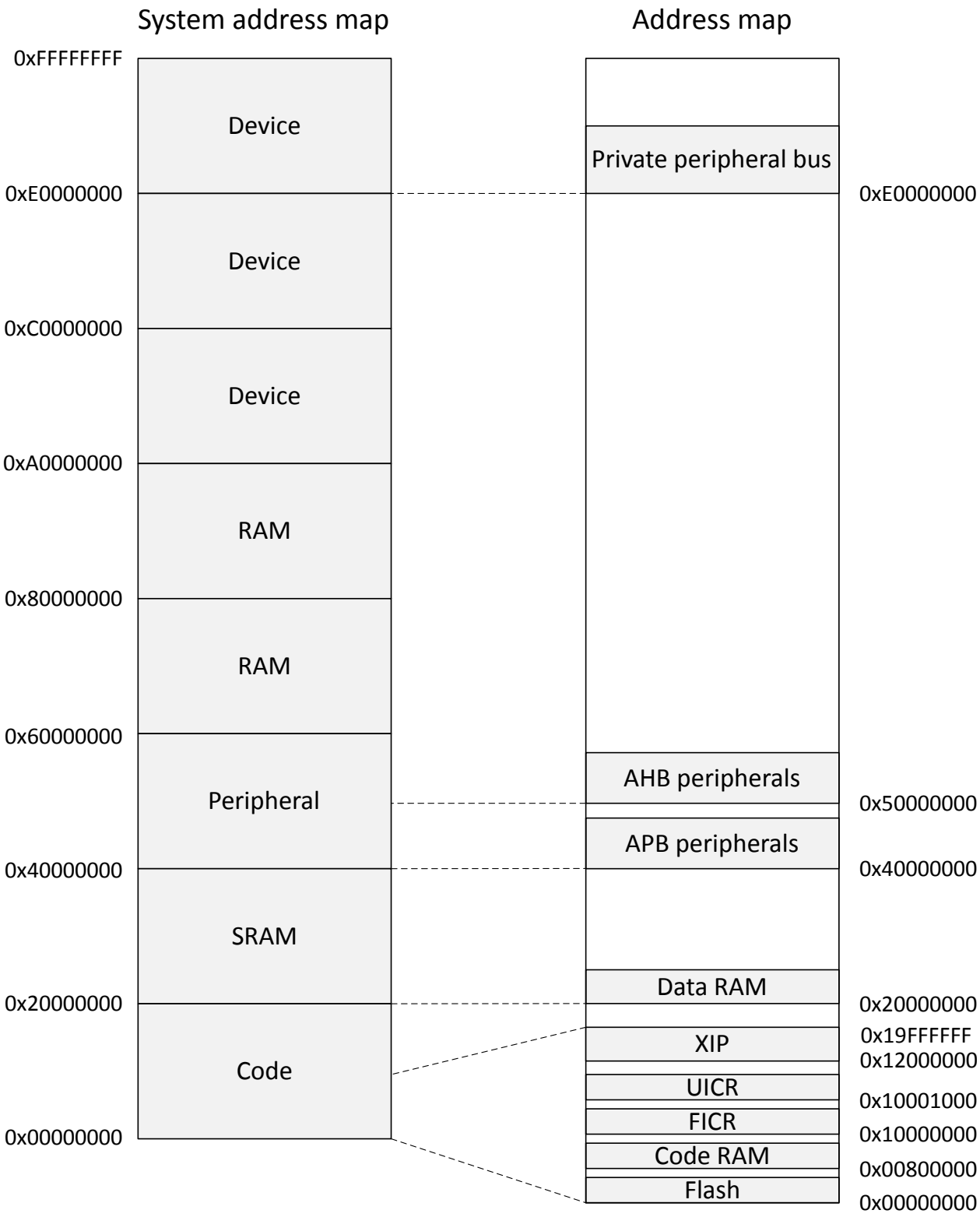

Figure 3: Memory map

4.2.4 Instantiation

ID	Base address	Peripheral	Instance	Description	
0	0x40000000	CLOCK	CLOCK	Clock control	
0	0x40000000	POWER	POWER	Power control	
1	0x40001000	RADIO	RADIO	2.4 GHz radio	
2	0x40002000	UART	UART0	Universal asynchronous receiver/transmitter	Deprecated
2	0x40002000	UARTE	UARTE0	Universal asynchronous receiver/transmitter with EasyDMA, unit 0	
3	0x40003000	SPI	SPI0	SPI master 0	Deprecated
3	0x40003000	SPIM	SPIM0	SPI master 0	
3	0x40003000	SPIS	SPIS0	SPI slave 0	
3	0x40003000	TWI	TWI0	Two-wire interface master 0	Deprecated
3	0x40003000	TWIM	TWIM0	Two-wire interface master 0	
3	0x40003000	TWIS	TWIS0	Two-wire interface slave 0	
4	0x40004000	SPI	SPI1	SPI master 1	Deprecated
4	0x40004000	SPIM	SPIM1	SPI master 1	
4	0x40004000	SPIS	SPIS1	SPI slave 1	
4	0x40004000	TWI	TWI1	Two-wire interface master 1	Deprecated
4	0x40004000	TWIM	TWIM1	Two-wire interface master 1	
4	0x40004000	TWIS	TWIS1	Two-wire interface slave 1	
5	0x40005000	NFCT	NFCT	Near field communication tag	
6	0x40006000	GPIOTE	GPIOTE	GPIO tasks and events	
7	0x40007000	SAADC	SAADC	Analog to digital converter	
8	0x40008000	TIMER	TIMER0	Timer 0	
9	0x40009000	TIMER	TIMER1	Timer 1	
10	0x4000A000	TIMER	TIMER2	Timer 2	
11	0x4000B000	RTC	RTC0	Real-time counter 0	
12	0x4000C000	TEMP	TEMP	Temperature sensor	
13	0x4000D000	RNG	RNG	Random number generator	
14	0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption	
15	0x4000F000	AAR	AAR	Accelerated address resolver	
15	0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption	
16	0x40010000	WDT	WDT	Watchdog timer	
17	0x40011000	RTC	RTC1	Real-time counter 1	
18	0x40012000	QDEC	QDEC	Quadrature decoder	
19	0x40013000	COMP	COMP	General purpose comparator	
19	0x40013000	LPCOMP	LPCOMP	Low power comparator	
20	0x40014000	EGU	EGU0	Event generator unit 0	
20	0x40014000	SWI	SWI0	Software interrupt 0	
21	0x40015000	EGU	EGU1	Event generator unit 1	
21	0x40015000	SWI	SWI1	Software interrupt 1	
22	0x40016000	EGU	EGU2	Event generator unit 2	
22	0x40016000	SWI	SWI2	Software interrupt 2	
23	0x40017000	EGU	EGU3	Event generator unit 3	
23	0x40017000	SWI	SWI3	Software interrupt 3	
24	0x40018000	EGU	EGU4	Event generator unit 4	
24	0x40018000	SWI	SWI4	Software interrupt 4	
25	0x40019000	EGU	EGU5	Event generator unit 5	
25	0x40019000	SWI	SWI5	Software interrupt 5	
26	0x4001A000	TIMER	TIMER3	Timer 3	
27	0x4001B000	TIMER	TIMER4	Timer 4	
28	0x4001C000	PWM	PWM0	Pulse width modulation unit 0	

ID	Base address	Peripheral	Instance	Description	
29	0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface	
30	0x4001E000	ACL	ACL	Access control lists	
30	0x4001E000	NVMC	NVMC	Non-volatile memory controller	
31	0x4001F000	PPI	PPI	Programmable peripheral interconnect	
32	0x40020000	MWU	MWU	Memory watch unit	
33	0x40021000	PWM	PWM1	Pulse width modulation unit 1	
34	0x40022000	PWM	PWM2	Pulse width modulation unit 2	
35	0x40023000	SPI	SPI2	SPI master 2	Deprecated
35	0x40023000	SPIM	SPIM2	SPI master 2	
35	0x40023000	SPIS	SPIS2	SPI slave 2	
36	0x40024000	RTC	RTC2	Real-time counter 2	
37	0x40025000	I2S	I2S	Inter-IC sound interface	
38	0x40026000	FPU	FPU	FPU interrupt	
39	0x40027000	USBD	USBD	Universal serial bus device	
40	0x40028000	UARTE	UARTE1	Universal asynchronous receiver/transmitter with EasyDMA, unit 1	
41	0x40029000	QSPI	QSPI	External memory interface	
45	0x4002D000	PWM	PWM3	Pulse width modulation unit 3	
47	0x4002F000	SPIM	SPIM3	SPI master 3	
0	0x50000000	GPIO	GPIO	General purpose input and output	Deprecated
0	0x50000000	GPIO	P0	General purpose input and output, port 0	
0	0x50000300	GPIO	P1	General purpose input and output, port 1	
42	0x5002A000	CRYPTOCELL	CRYPTOCELL	CryptoCell subsystem control interface	
N/A	0x10000000	FICR	FICR	Factory information configuration	
N/A	0x10001000	UICR	UICR	User information configuration	

Table 3: Instantiation table

4.3 NVMC — Non-volatile memory controller

The non-volatile memory controller (NVMC) is used for writing and erasing of the internal flash memory and the UICR (user information configuration registers).

The CONFIG on page 26 is used to enable the NVMC for writing (CONFIG.WEN = Wen) and erasing (CONFIG.WEN = Een). The user must make sure that writing and erasing are not enabled at the same time. Having both enabled at the same time may result in unpredictable behavior.

4.3.1 Writing to flash

When write is enabled, full 32-bit words can be written to word-aligned addresses in the flash.

As illustrated in Memory on page 19, the flash is divided into multiple pages. The same 32-bit word in the flash can only be written n_{WRITE} number of times before a page erase must be performed.

The NVMC is only able to write `0` to bits in the flash that are erased (set to `1`). It cannot rewrite a bit back to `1`. Only full 32-bit words can be written to flash using the NVMC interface. To write less than 32 bits, write the data as a full 32-bit word and set all the bits that should remain unchanged in the word to `1`. Note that the restriction on the number of writes (n_{WRITE}) still applies in this case.

Only word-aligned writes are allowed. Byte or half-word-aligned writes will result in a hard fault.

The time it takes to write a word to flash is specified by t_{WRITE}. The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the flash.

NVM writing time can be reduced by using READYNEXT. If this status bit is set to '1', code can perform the next data write to the flash. This write will be buffered and will be taken into account as soon as the ongoing write operation is completed.

4.3.2 Erasing a page in flash

When erase is enabled, the flash memory can be erased page by page using the ERASEPAGE on page 26.

After erasing a flash page, all bits in the page are set to `1`. The time it takes to erase a page is specified by $t_{ERASEPAGE}$. The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the flash.

See Partial erase of a page in flash on page 24 for information on dividing the page erase time into shorter chunks.

4.3.3 Writing to user information configuration registers (UICR)

User information configuration registers (UICR) are written in the same way as flash. After UICR has been written, the new UICR configuration will only take effect after a reset.

UICR can only be written n_{WRITE} number of times before an erase must be performed using ERASEUICR on page 28 or ERASEALL on page 27. The time it takes to write a word to UICR is specified by t_{WRITE}.
The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the UICR.

4.3.4 Erasing user information configuration registers (UICR)

When erase is enabled, UICR can be erased using the ERASEUICR on page 28.

After erasing UICR all bits in UICR are set to `1`. The time it takes to erase UICR is specified by $t_{ERASEPAGE}$. The CPU is halted if the CPU executes code from the flash while the NVMC performs the erase operation.

4.3.5 Erase all

When erase is enabled, flash and UICR can be erased completely in one operation by using the ERASEALL on page 27. This operation will not erase the factory information configuration registers (FICR).

The time it takes to perform an ERASEALL command is specified by $t_{ERASEALL}$ The CPU is halted if the CPU executes code from the flash while the NVMC performs the erase operation.

4.3.6 Access port protection behavior

When access port protection is enabled, parts of the NVMC functionality will be blocked in order to prevent intentional or unintentional erase of UICR.

	CTRL-AP ERASEALL	NVMC ERASEPAGE	NVMC ERASEPAGE PARTIAL	NVMC ERASEALL	NVMC ERASEUICR
APPROTECT					
Disabled	Allowed	Allowed	Allowed	Allowed	Allowed
Enabled	Allowed	Allowed	Allowed	Allowed	Blocked

Table 4: NVMC Protection

4.3.7 Partial erase of a page in flash

Partial erase is a feature in the NVMC to split a page erase time into shorter chunks, so this can be used to prevent longer CPU stalls in time-critical applications. Partial erase is only applicable to the code area in the flash and does not work with UICR.

When erase is enabled, the partial erase of a flash page can be started by writing to ERASEPAGEPARTIAL on page 28. The duration of a partial erase can be configured in ERASEPAGEPARTIALCFG on page 28. A flash page is erased when its erase time reaches $t_{ERASEPAGE}$. Use ERASEPAGEPARTIAL N number of times so that N * ERASEPAGEPARTIALCFG ≥ $t_{ERASEPAGE}$, where N * ERASEPAGEPARTIALCFG gives the cumulative (total) erase time. Every time the cumulative erase time reaches $t_{ERASEPAGE}$, it counts as one erase cycle.

After the erase is done, all bits in the page are set to '1'. The CPU is halted if the CPU executes code from the flash while the NVMC performs the partial erase operation.

The bits in the page are undefined if the flash page erase is incomplete, i.e. if a partial erase has started but the total erase time is less than $t_{ERASEPAGE}$.

4.3.8 Cache

An instruction cache (I-Cache) can be enabled for the ICODE bus in the NVMC.

See the Memory map in Memory map on page 20 for the location of flash.

A cache hit is an instruction fetch from the cache, and it has a 0 wait-state delay. The number of wait-states for a cache miss, where the instruction is not available in the cache and needs to be fetched from flash, depends on the processor frequency and is shown in CPU on page 18

Enabling the cache can increase CPU performance and reduce power consumption by reducing the number of wait cycles and the number of flash accesses. This will depend on the cache hit rate. Cache will use some current when enabled. If the reduction in average current due to reduced flash accesses is larger than the cache power requirement, the average current to execute the program code will reduce.

When disabled, the cache does not use current and does not retain its content.

It is possible to enable cache profiling to analyze the performance of the cache for your program using the ICACHECNF register. When profiling is enabled, the IHIT and IMISS registers are incremented for every instruction cache hit or miss respectively. The hit and miss profiling registers do not wrap around after reaching the maximum value. If the maximum value is reached, consider profiling for a shorter duration to get correct numbers.

4.3.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001E000	NVMC	NVMC	Non-volatile memory controller	

Table 5: Instances

Register	Offset	Description	
READY	0x400	Ready flag	
READYNEXT	0x408	Ready flag	
CONFIG	0x504	Configuration register	
ERASEPAGE	0x508	Register for erasing a page in code area	
ERASEPCR1	0x508	Register for erasing a page in code area. Equivalent to ERASEPAGE.	Deprecated
ERASEALL	0x50C	Register for erasing all non-volatile user memory	
ERASEPCR0	0x510	Register for erasing a page in code area. Equivalent to ERASEPAGE.	Deprecated
ERASEUICR	0x514	Register for erasing user information configuration registers	
ERASEPAGEPARTIAL	0x518	Register for partial erase of a page in code area	
ERASEPAGEPARTIALCFG	0x51C	Register for partial erase configuration	
ICACHECNF	0x540	I-code cache configuration register.	
IHIT	0x548	I-code cache hit counter.	

Register	Offset	Description
IMISS	0x54C	I-code cache miss counter.

Table 6: Register overview

4.3.9.1 READY

Address offset: 0x400

Ready flag

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	R	READY			NVMC is ready or busy
			Busy	0	NVMC is busy (on-going write or erase operation)
			Ready	1	NVMC is ready

4.3.9.2 READYNEXT

Address offset: 0x408

Ready flag

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	READYNEXT			NVMC can accept a new write operation
			Busy	0	NVMC cannot accept any write operation
			Ready	1	NVMC is ready

4.3.9.3 CONFIG

Address offset: 0x504

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	WEN			Program memory access mode. It is strongly recommended to only activate erase and write modes when they are actively used. Enabling write or erase will invalidate the cache and keep it invalidated.
			Ren	0	Read only access
			Wen	1	Write enabled
			Een	2	Erase enabled

4.3.9.4 ERASEPAGE

Address offset: 0x508

Register for erasing a page in code area

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEPAGE			Register for starting erase of a page in code area The value is the address to the page to be erased. (Addresses of first word in page). Note that the erase must be enabled using CONFIG.WEN before the page can be erased. Attempts to erase pages that are outside the code area may result in undesirable behaviour, e.g. the wrong page may be erased.

4.3.9.5 ERASEPCR1 (Deprecated)

Address offset: 0x508

Register for erasing a page in code area. Equivalent to ERASEPAGE.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEPCR1			Register for erasing a page in code area. Equivalent to ERASEPAGE.

4.3.9.6 ERASEALL

Address offset: 0x50C

Register for erasing all non-volatile user memory

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEALL			Erase all non-volatile memory including UICR registers. Note that the erase must be enabled using CONFIG.WEN before the non-volatile memory can be erased.
			NoOperation	0	No operation
			Erase	1	Start chip erase

4.3.9.7 ERASEPCR0 (Deprecated)

Address offset: 0x510

Register for erasing a page in code area. Equivalent to ERASEPAGE.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEPCR0			Register for starting erase of a page in code area. Equivalent to ERASEPAGE.

4.3.9.8 ERASEUICR

Address offset: 0x514

Register for erasing user information configuration registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEUICR			Register starting erase of all user information configuration registers. Note that the erase must be enabled using CONFIG.WEN before the UICR can be erased.
			NoOperation	0	No operation
			Erase	1	Start erase of UICR

4.3.9.9 ERASEPAGEPARTIAL

Address offset: 0x518

Register for partial erase of a page in code area

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ERASEPAGEPARTIAL			Register for starting partial erase of a page in code area The value is the address to the page to be partially erased (address of the first word in page). Note that the erase must be enabled using CONFIG.WEN before every erase page partial and disabled using CONFIG.WEN after every erase page partial. Attempts to erase pages that are outside the code area may result in undesirable behaviour, e.g. the wrong page may be erased.

4.3.9.10 ERASEPAGEPARTIALCFG

Address offset: 0x51C

Register for partial erase configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x0000000A				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DURATION			Duration of the partial erase in milliseconds The user must ensure that the total erase time is long enough for a complete erase of the flash page.

4.3.9.11 ICACHECNF

Address offset: 0x540

I-code cache configuration register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CACHEEN			Cache enable
			Disabled	0	Disable cache. Invalidates all cache entries.
			Enabled	1	Enable cache
B	RW	CACHEPROFEN			Cache profiling enable
			Disabled	0	Disable cache profiling
			Enabled	1	Enable cache profiling

4.3.9.12 IHIT

Address offset: 0x548

I-code cache hit counter.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	HITS			Number of cache hits

4.3.9.13 IMISS

Address offset: 0x54C

I-code cache miss counter.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MISSES			Number of cache misses

4.3.10 Electrical specification

4.3.10.1 Flash programming

Symbol	Description	Min.	Typ.	Max.	Units
n_{WRITE}	Number of times a 32-bit word can be written before erase			2	
$n_{ENDURANCE}$	Erase cycles per page	10000			
t_{WRITE}	Time to write one 32-bit word			41[1]	µs
$t_{ERASEPAGE}$	Time to erase one page			85[1]	ms
$t_{ERASEALL}$	Time to erase all flash			169[1]	ms
$t_{ERASEPAGEPARTIAL,acc}$	Accuracy of the partial page erase duration. Total execution time for one partial page erase is defined as ERASEPAGEPARTIALCFG * $t_{ERASEPAGEPARTIAL,acc}$.			1.05[1]	

[1] Applies when HFXO is used. Timing varies according to HFINT accuracy when HFINT is used.

4.3.10.2 Cache size

Symbol	Description	Min.	Typ.	Max.	Units
$Size_{ICODE}$	I-Code cache size		2048		Bytes

4.4 FICR — Factory information configuration registers

Factory information configuration registers (FICR) are pre-programmed in factory and cannot be erased by the user. These registers contain chip-specific information and configuration.

4.4.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x10000000	FICR	FICR	Factory information configuration	

Table 7: Instances

Register	Offset	Description
CODEPAGESIZE	0x010	Code memory page size
CODESIZE	0x014	Code memory size
DEVICEID[0]	0x060	Device identifier
DEVICEID[1]	0x064	Device identifier
ER[0]	0x080	Encryption root, word 0
ER[1]	0x084	Encryption root, word 1
ER[2]	0x088	Encryption root, word 2
ER[3]	0x08C	Encryption root, word 3
IR[0]	0x090	Identity Root, word 0
IR[1]	0x094	Identity Root, word 1
IR[2]	0x098	Identity Root, word 2
IR[3]	0x09C	Identity Root, word 3
DEVICEADDRTYPE	0x0A0	Device address type
DEVICEADDR[0]	0x0A4	Device address 0
DEVICEADDR[1]	0x0A8	Device address 1
INFO.PART	0x100	Part code
INFO.VARIANT	0x104	Build code (hardware version and production configuration)
INFO.PACKAGE	0x108	Package option
INFO.RAM	0x10C	RAM variant
INFO.FLASH	0x110	Flash variant
INFO.UNUSED8[0]	0x114	Reserved
INFO.UNUSED8[1]	0x118	Reserved
INFO.UNUSED8[2]	0x11C	Reserved
PRODTEST[0]	0x350	Production test signature 0
PRODTEST[1]	0x354	Production test signature 1
PRODTEST[2]	0x358	Production test signature 2
TEMP.A0	0x404	Slope definition A0
TEMP.A1	0x408	Slope definition A1
TEMP.A2	0x40C	Slope definition A2
TEMP.A3	0x410	Slope definition A3
TEMP.A4	0x414	Slope definition A4
TEMP.A5	0x418	Slope definition A5
TEMP.B0	0x41C	Y-intercept B0

Register	Offset	Description
TEMP.B1	0x420	Y-intercept B1
TEMP.B2	0x424	Y-intercept B2
TEMP.B3	0x428	Y-intercept B3
TEMP.B4	0x42C	Y-intercept B4
TEMP.B5	0x430	Y-intercept B5
TEMP.T0	0x434	Segment end T0
TEMP.T1	0x438	Segment end T1
TEMP.T2	0x43C	Segment end T2
TEMP.T3	0x440	Segment end T3
TEMP.T4	0x444	Segment end T4
NFC.TAGHEADER0	0x450	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER1	0x454	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER2	0x458	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER3	0x45C	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
TRNG90B.BYTES	0xC00	Amount of bytes for the required entropy bits
TRNG90B.RCCUTOFF	0xC04	Repetition counter cutoff
TRNG90B.APCUTOFF	0xC08	Adaptive proportion cutoff
TRNG90B.STARTUP	0xC0C	Amount of bytes for the startup tests
TRNG90B.ROSC1	0xC10	Sample count for ring oscillator 1
TRNG90B.ROSC2	0xC14	Sample count for ring oscillator 2
TRNG90B.ROSC3	0xC18	Sample count for ring oscillator 3
TRNG90B.ROSC4	0xC1C	Sample count for ring oscillator 4

Table 8: Register overview

4.4.1.1 CODEPAGESIZE

Address offset: 0x010

Code memory page size

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	CODEPAGESIZE			Code memory page size

4.4.1.2 CODESIZE

Address offset: 0x014

Code memory size

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	CODESIZE			Code memory size in number of pages Total code space is: CODEPAGESIZE * CODESIZE

4.4.1.3 DEVICEID[n] (n=0..1)

Address offset: 0x060 + (n × 0x4)

Device identifier

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	DEVICEID			64 bit unique device identifier DEVICEID[0] contains the least significant bits of the device identifier. DEVICEID[1] contains the most significant bits of the device identifier.

4.4.1.4 ER[n] (n=0..3)

Address offset: 0x080 + (n × 0x4)

Encryption root, word n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	ER			Encryption root, word n

4.4.1.5 IR[n] (n=0..3)

Address offset: 0x090 + (n × 0x4)

Identity Root, word n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	IR			Identity Root, word n

4.4.1.6 DEVICEADDRTYPE

Address offset: 0x0A0

Device address type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	DEVICEADDRTYPE			Device address type
			Public	0	Public address
			Random	1	Random address

4.4.1.7 DEVICEADDR[n] (n=0..1)

Address offset: 0x0A4 + (n × 0x4)

Device address n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	DEVICEADDR			48 bit device address DEVICEADDR[0] contains the least significant bits of the device address. DEVICEADDR[1] contains the most significant bits of the device address. Only bits [15:0] of DEVICEADDR[1] are used.

4.4.1.8 INFO.PART

Address offset: 0x100

Part code

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00052840				**0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	PART			Part code
			N52840	0x52840	nRF52840
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.9 INFO.VARIANT

Address offset: 0x104

Build code (hardware version and production configuration)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	VARIANT			Build code (hardware version and production configuration). Encoded as ASCII.
			AAAA	0x41414141	AAAA
			BAAA	0x42414141	BAAA
			CAAA	0x43414141	CAAA
			AABA	0x41414241	AABA
			AABB	0x41414242	AABB
			AACA	0x41414341	AACA
			AAAB	0x41414142	AAAB
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.10 INFO.PACKAGE

Address offset: 0x108

Package option

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	PACKAGE			Package option
			QI	0x2004	QIxx - 73-pin aQFN
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.11 INFO.RAM

Address offset: 0x10C

RAM variant

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	RAM			RAM variant
			K16	0x10	16 kByte RAM
			K32	0x20	32 kByte RAM
			K64	0x40	64 kByte RAM
			K128	0x80	128 kByte RAM
			K256	0x100	256 kByte RAM
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.12 INFO.FLASH

Address offset: 0x110

Flash variant

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	FLASH			Flash variant
			K128	0x80	128 kByte FLASH
			K256	0x100	256 kByte FLASH
			K512	0x200	512 kByte FLASH
			K1024	0x400	1 MByte FLASH
			K2048	0x800	2 MByte FLASH
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.13 PRODTEST[n] (n=0..2)

Address offset: 0x350 + (n × 0x4)

Production test signature n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	PRODTEST			Production test signature n
			Done	0xBB42319F	Production tests done
			NotDone	0xFFFFFFFF	Production tests not done

4.4.1.14 TEMP.A0

Address offset: 0x404

Slope definition A0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFF320				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.15 TEMP.A1

Address offset: 0x408

Slope definition A1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFF343				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 0 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.16 TEMP.A2

Address offset: 0x40C

Slope definition A2

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFF35D				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0 1**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.17 TEMP.A3

Address offset: 0x410

Slope definition A3

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0xFFFFF400				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.18 TEMP.A4

Address offset: 0x414

Slope definition A4

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0xFFFFF452				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.19 TEMP.A5

Address offset: 0x418

Slope definition A5

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0xFFFFF37B				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1**
ID	RW	Field	Value ID	Value	Description
A	R	A			A (slope definition) register.

4.4.1.20 TEMP.B0

Address offset: 0x41C

Y-intercept B0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0xFFFF3FCC				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.21 TEMP.B1

Address offset: 0x420

Y-intercept B1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0xFFFF3F98				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.22 TEMP.B2

Address offset: 0x424

Y-intercept B2

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFF3F98	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.23 TEMP.B3

Address offset: 0x428

Y-intercept B3

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFF0012	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.24 TEMP.B4

Address offset: 0x42C

Y-intercept B4

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFF004D	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.25 TEMP.B5

Address offset: 0x430

Y-intercept B5

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFF3E10	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	B			B (y-intercept)

4.4.1.26 TEMP.T0

Address offset: 0x434

Segment end T0

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0xFFFFFFE2	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0**

ID	RW	Field	Value ID	Value	Description
A	R	T			T (segment end) register

4.4.1.27 TEMP.T1

Address offset: 0x438

Segment end T1

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0xFFFFFF00	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	T			T (segment end) register

4.4.1.28 TEMP.T2

Address offset: 0x43C

Segment end T2

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0xFFFFFF14	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	T			T (segment end) register

4.4.1.29 TEMP.T3

Address offset: 0x440

Segment end T3

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0xFFFFFF19	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1**

ID	RW	Field	Value ID	Value	Description
A	R	T			T (segment end) register

4.4.1.30 TEMP.T4

Address offset: 0x444

Segment end T4

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0xFFFFFF50	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	T			T (segment end) register

4.4.1.31 NFC.TAGHEADER0

Address offset: 0x450

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.

Bit number																																
ID																																
Reset 0xFFFFFF5F																																
ID	RW	Field	Value ID	Value	Description																											
A	R	MFGID			Default Manufacturer ID: Nordic Semiconductor ASA has ICM 0x5F																											
B	R	UD1			Unique identifier byte 1																											
C	R	UD2			Unique identifier byte 2																											
D	R	UD3			Unique identifier byte 3																											

4.4.1.32 NFC.TAGHEADER1

Address offset: 0x454

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	RW	Field	Value ID	Value	Description																											
A-D	R	UD[i] (i=4..7)			Unique identifier byte i																											

4.4.1.33 NFC.TAGHEADER2

Address offset: 0x458

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	RW	Field	Value ID	Value	Description																											
A-D	R	UD[i] (i=8..11)			Unique identifier byte i																											

4.4.1.34 NFC.TAGHEADER3

Address offset: 0x45C

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	RW	Field	Value ID	Value	Description																											
A-D	R	UD[i] (i=12..15)			Unique identifier byte i																											

4.4.1.35 TRNG90B.BYTES

Address offset: 0xC00

Amount of bytes for the required entropy bits

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	BYTES			Amount of bytes for the required entropy bits

4.4.1.36 TRNG90B.RCCUTOFF

Address offset: 0xC04

Repetition counter cutoff

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	RCCUTOFF			Repetition counter cutoff

4.4.1.37 TRNG90B.APCUTOFF

Address offset: 0xC08

Adaptive proportion cutoff

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	R	APCUTOFF			Adaptive proportion cutoff

4.4.1.38 TRNG90B.STARTUP

Address offset: 0xC0C

Amount of bytes for the startup tests

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000210	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	STARTUP			Amount of bytes for the startup tests

4.4.1.39 TRNG90B.ROSC1

Address offset: 0xC10

Sample count for ring oscillator 1

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	R	ROSC1			Sample count for ring oscillator 1

4.4.1.40 TRNG90B.ROSC2

Address offset: 0xC14

Sample count for ring oscillator 2

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	R	ROSC2			Sample count for ring oscillator 2

4.4.1.41 TRNG90B.ROSC3

Address offset: 0xC18

Sample count for ring oscillator 3

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	R	ROSC3			Sample count for ring oscillator 3

4.4.1.42 TRNG90B.ROSC4

Address offset: 0xC1C

Sample count for ring oscillator 4

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	R	ROSC4			Sample count for ring oscillator 4

4.5 UICR — User information configuration registers

The user information configuration registers (UICRs) are non-volatile memory (NVM) registers for configuring user-specific settings.

For information on writing UICR registers, see the NVMC — Non-volatile memory controller on page 23 and Memory on page 19 chapters.

4.5.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x10001000	UICR	UICR	User information configuration	

Table 9: Instances

Register	Offset	Description	
UNUSED0	0x000		Reserved
UNUSED1	0x004		Reserved
UNUSED2	0x008		Reserved
UNUSED3	0x010		Reserved
NRFFW[0]	0x014	Reserved for Nordic firmware design	
NRFFW[1]	0x018	Reserved for Nordic firmware design	
NRFFW[2]	0x01C	Reserved for Nordic firmware design	
NRFFW[3]	0x020	Reserved for Nordic firmware design	
NRFFW[4]	0x024	Reserved for Nordic firmware design	
NRFFW[5]	0x028	Reserved for Nordic firmware design	
NRFFW[6]	0x02C	Reserved for Nordic firmware design	
NRFFW[7]	0x030	Reserved for Nordic firmware design	
NRFFW[8]	0x034	Reserved for Nordic firmware design	
NRFFW[9]	0x038	Reserved for Nordic firmware design	
NRFFW[10]	0x03C	Reserved for Nordic firmware design	
NRFFW[11]	0x040	Reserved for Nordic firmware design	
NRFFW[12]	0x044	Reserved for Nordic firmware design	
NRFFW[13]	0x048	Reserved for Nordic firmware design	
NRFFW[14]	0x04C	Reserved for Nordic firmware design	
NRFHW[0]	0x050	Reserved for Nordic hardware design	
NRFHW[1]	0x054	Reserved for Nordic hardware design	
NRFHW[2]	0x058	Reserved for Nordic hardware design	
NRFHW[3]	0x05C	Reserved for Nordic hardware design	
NRFHW[4]	0x060	Reserved for Nordic hardware design	
NRFHW[5]	0x064	Reserved for Nordic hardware design	
NRFHW[6]	0x068	Reserved for Nordic hardware design	
NRFHW[7]	0x06C	Reserved for Nordic hardware design	
NRFHW[8]	0x070	Reserved for Nordic hardware design	
NRFHW[9]	0x074	Reserved for Nordic hardware design	
NRFHW[10]	0x078	Reserved for Nordic hardware design	
NRFHW[11]	0x07C	Reserved for Nordic hardware design	
CUSTOMER[0]	0x080	Reserved for customer	
CUSTOMER[1]	0x084	Reserved for customer	
CUSTOMER[2]	0x088	Reserved for customer	
CUSTOMER[3]	0x08C	Reserved for customer	
CUSTOMER[4]	0x090	Reserved for customer	
CUSTOMER[5]	0x094	Reserved for customer	
CUSTOMER[6]	0x098	Reserved for customer	
CUSTOMER[7]	0x09C	Reserved for customer	
CUSTOMER[8]	0x0A0	Reserved for customer	
CUSTOMER[9]	0x0A4	Reserved for customer	
CUSTOMER[10]	0x0A8	Reserved for customer	
CUSTOMER[11]	0x0AC	Reserved for customer	
CUSTOMER[12]	0x0B0	Reserved for customer	
CUSTOMER[13]	0x0B4	Reserved for customer	

Register	Offset	Description
CUSTOMER[14]	0x0B8	Reserved for customer
CUSTOMER[15]	0x0BC	Reserved for customer
CUSTOMER[16]	0x0C0	Reserved for customer
CUSTOMER[17]	0x0C4	Reserved for customer
CUSTOMER[18]	0x0C8	Reserved for customer
CUSTOMER[19]	0x0CC	Reserved for customer
CUSTOMER[20]	0x0D0	Reserved for customer
CUSTOMER[21]	0x0D4	Reserved for customer
CUSTOMER[22]	0x0D8	Reserved for customer
CUSTOMER[23]	0x0DC	Reserved for customer
CUSTOMER[24]	0x0E0	Reserved for customer
CUSTOMER[25]	0x0E4	Reserved for customer
CUSTOMER[26]	0x0E8	Reserved for customer
CUSTOMER[27]	0x0EC	Reserved for customer
CUSTOMER[28]	0x0F0	Reserved for customer
CUSTOMER[29]	0x0F4	Reserved for customer
CUSTOMER[30]	0x0F8	Reserved for customer
CUSTOMER[31]	0x0FC	Reserved for customer
PSELRESET[0]	0x200	Mapping of the nRESET function
PSELRESET[1]	0x204	Mapping of the nRESET function
APPROTECT	0x208	Access port protection
NFCPINS	0x20C	Setting of pins dedicated to NFC functionality: NFC antenna or GPIO
DEBUGCTRL	0x210	Processor debug control
REGOUT0	0x304	GPIO reference voltage / external output supply voltage in high voltage mode

Table 10: Register overview

4.5.1.1 NRFFW[n] (n=0..14)

Address offset: 0x014 + (n × 0x4)

Reserved for Nordic firmware design

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	RW	NRFFW			Reserved for Nordic firmware design

4.5.1.2 NRFHW[n] (n=0..11)

Address offset: 0x050 + (n × 0x4)

Reserved for Nordic hardware design

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	RW	NRFHW			Reserved for Nordic hardware design

4.5.1.3 CUSTOMER[n] (n=0..31)

Address offset: 0x080 + (n × 0x4)

Reserved for customer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	CUSTOMER			Reserved for customer

4.5.1.4 PSELRESET[n] (n=0..1)

Address offset: 0x200 + (n × 0x4)

Mapping of the nRESET function

All PSELRESET registers have to contain the same value for a pin mapping to be valid. If they do not, there will be no nRESET function exposed on a GPIO, and the device will always start independently of the levels present on any of the GPIOs.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		18	Pin number of PORT onto which nRESET is exposed
B	RW	PORT		0	Port number onto which nRESET is exposed
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

4.5.1.5 APPROTECT

Address offset: 0x208

Access port protection

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PALL			Enable or disable access port protection. See Debug and trace on page 49 for more information.
			Disabled	0xFF	Disable
			Enabled	0x00	Enable

4.5.1.6 NFCPINS

Address offset: 0x20C

Setting of pins dedicated to NFC functionality: NFC antenna or GPIO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PROTECT			Setting of pins dedicated to NFC functionality
			Disabled	0	Operation as GPIO pins. Same protection as normal GPIO pins
			NFC	1	Operation as NFC antenna pins. Configures the protection for NFC operation

4.5.1.7 DEBUGCTRL

Address offset: 0x210

Processor debug control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	CPUNIDEN			Configure CPU non-intrusive debug features
			Enabled	0xFF	Enable CPU ITM and ETM functionality (default behavior)
			Disabled	0x00	Disable CPU ITM and ETM functionality
B	RW	CPUFPBEN			Configure CPU flash patch and breakpoint (FPB) unit behavior
			Enabled	0xFF	Enable CPU FPB unit (default behavior)
			Disabled	0x00	Disable CPU FPB unit. Writes into the FPB registers will be ignored.

4.5.1.8 REGOUT0

Address offset: 0x304

GPIO reference voltage / external output supply voltage in high voltage mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	VOUT			Output voltage from of REG0 regulator stage. The maximum output voltage from this stage is given as VDDH - VEXDIF.
			1V8	0	1.8 V
			2V1	1	2.1 V
			2V4	2	2.4 V
			2V7	3	2.7 V
			3V0	4	3.0 V
			3V3	5	3.3 V
			DEFAULT	7	Default voltage: 1.8 V

4.6 EasyDMA

EasyDMA is a module implemented by some peripherals to gain direct access to Data RAM.

EasyDMA is an AHB bus master similar to CPU and is connected to the AHB multilayer interconnect for direct access to Data RAM. EasyDMA is not able to access flash.

A peripheral can implement multiple EasyDMA instances to provide dedicated channels. For example, for reading and writing of data between the peripheral and RAM. This concept is illustrated in EasyDMA example on page 46.

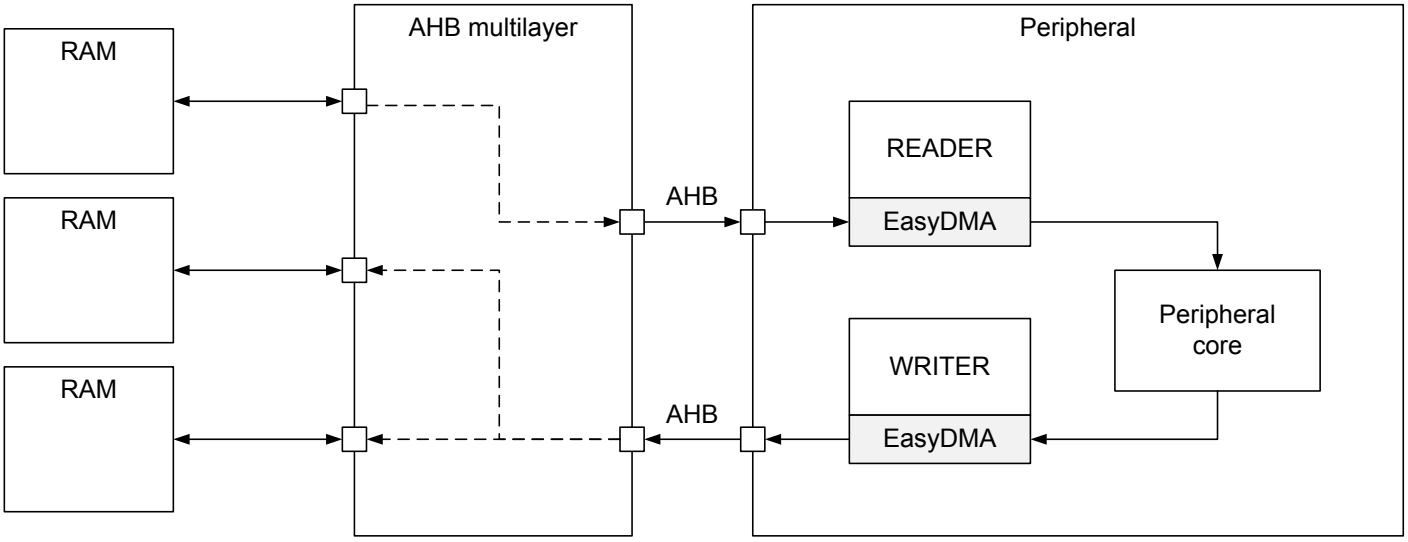

Figure 4: EasyDMA example

An EasyDMA channel is usually implemented like illustrated by the code below, but some variations may occur:

```
READERBUFFER_SIZE 5
WRITERBUFFER_SIZE 6

uint8_t readerBuffer[READERBUFFER_SIZE]  __at__ 0x20000000;
uint8_t writerBuffer[WRITERBUFFER_SIZE]  __at__ 0x20000005;

// Configuring the READER channel
MYPERIPHERAL->READER.MAXCNT = READERBUFFER_SIZE;
MYPERIPHERAL->READER.PTR = &readerBuffer;

// Configure the WRITER channel
MYPERIPHERAL->WRITER.MAXCNT = WRITEERBUFFER_SIZE;
MYPERIPHERAL->WRITER.PTR = &writerBuffer;
```

This example shows a peripheral called MYPERIPHERAL that implements two EasyDMA channels - one for reading called READER, and one for writing called WRITER. When the peripheral is started, it is assumed that the peripheral will:

- Read 5 bytes from the readerBuffer located in RAM at address 0x20000000.
- Process the data.
- Write no more than 6 bytes back to the writerBuffer located in RAM at address 0x20000005.

The memory layout of these buffers is illustrated in EasyDMA memory layout on page 47.

0x20000000	readerBuffer[0]	readerBuffer[1]	readerBuffer[2]	readerBuffer[3]
0x20000004	readerBuffer[4]	writerBuffer[0]	writerBuffer[1]	writerBuffer[2]
0x20000008	writerBuffer[3]	writerBuffer[4]	writerBuffer[5]	

Figure 5: EasyDMA memory layout

The WRITER.MAXCNT register should not be specified larger than the actual size of the buffer (writerBuffer). Otherwise, the channel would overflow the writerBuffer.

Once an EasyDMA transfer is completed, the AMOUNT register can be read by the CPU to see how many bytes were transferred. For example, CPU can read MYPERIPHERAL->WRITER.AMOUNT register to see how many bytes WRITER wrote to RAM.

4.6.1 EasyDMA array list

EasyDMA is able to operate in a mode called array list.

The array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

The EasyDMA array list can be implemented by using the data structure ArrayList_type as illustrated in the code example below:

```
#define BUFFER_SIZE  4

typedef struct ArrayList
{
  uint8_t buffer[BUFFER_SIZE];
} ArrayList_type;

ArrayList_type ReaderList[3];

READER.MAXCNT = BUFFER_SIZE;
READER.PTR = &ReaderList;
```

The data structure only includes a buffer with size equal to the size of READER.MAXCNT register. EasyDMA uses the READER.MAXCNT register to determine when the buffer is full.

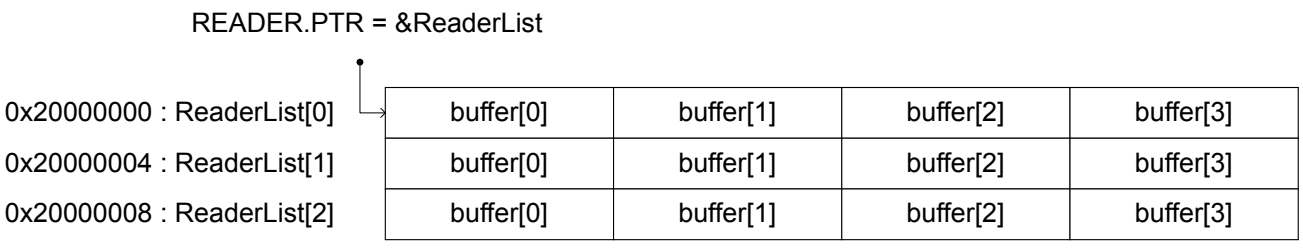

Figure 6: EasyDMA array list

4.7 AHB multilayer

AHB multilayer enables parallel access paths between multiple masters and slaves in a system. Access is resolved using priorities.

Each bus master is connected to the slave devices using an interconnection matrix. The bus masters are assigned priorities. Priorities are used to resolve access when two (or more) bus masters request access to the same slave device. The following applies:

- If two (or more) bus masters request access to the same slave device, the master with the highest priority is granted the access first.
- Bus masters with lower priority are stalled until the higher priority master has completed its transaction.
- If the higher priority master pauses at any point during its transaction, the lower priority master in queue is temporarily granted access to the slave device until the higher priority master resumes its activity.
- Bus masters that have the same priority are mutually exclusive, thus cannot be used concurrently.

Some peripherals, for example radio, do not have a safe stalling mechanism (no internal data buffering, nor opportunity to pause incoming data). Being a low priority bus master might cause loss of data for such peripherals upon bus contention. To avoid AHB bus contention when using multiple bus masters, apply one of the following guidelines:

- As a good general rule, avoid situations where more than one bus master is accessing the same slave.
- If more than one bus master is accessing the same slave, make sure that the bus bandwidth is not exhausted.

Below is a list of bus masters in the system and their priorities.

Bus master name	Description
CPU	
CTRL-AP	
USB	
CRYPTOCELL	
SPIM1/SPIS1/TWIM1/TWIS1	Same priority and mutually exclusive
RADIO	
CCM/ECB/AAR	Same priority and mutually exclusive
SAADC	
UARTE0	
SPIM0/SPIS0/TWIM0/TWIS0	Same priority and mutually exclusive
SPIM2/SPIS2	Same priority and mutually exclusive
NFCT	
I2S	
PDM	
PWM0	
PWM1	
PWM2	
QSPI	
PWM3	
UARTE1	
SPIM3	

Table 11: AHB bus masters (listed in priority order, highest to lowest)

Defined bus masters are the CPU and the peripherals with implemented EasyDMA, and the available slaves are RAM AHB slaves. How the bus masters and slaves are connected using the interconnection matrix is illustrated in Memory on page 19.

4.8 Debug and trace

Debug and trace system offers a flexible and powerful mechanism for non-intrusive debugging.

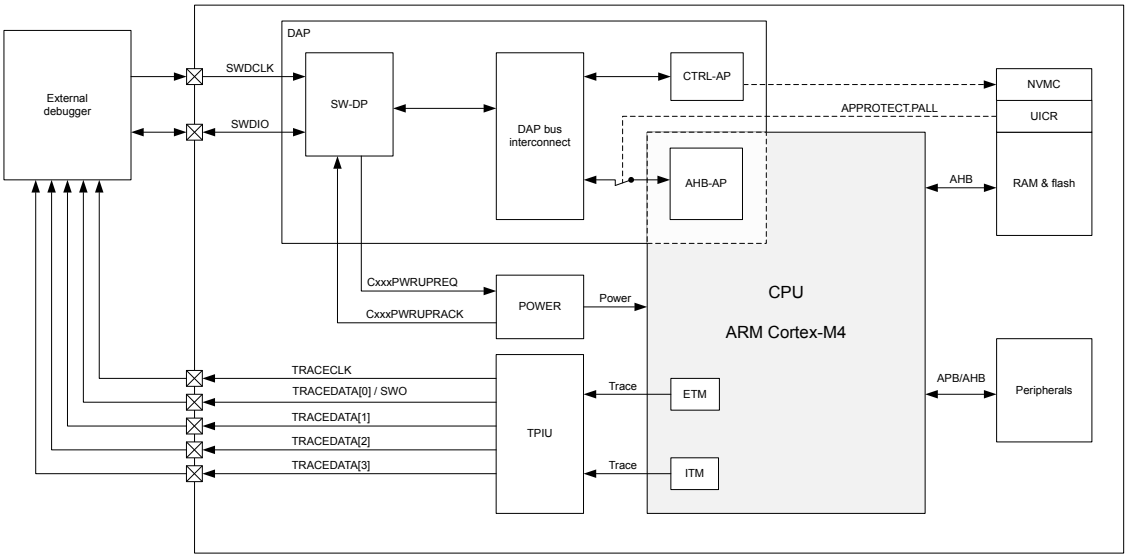

Figure 7: Overview

The main features of the debug and trace system are:

- Two-pin serial wire debug (SWD) interface
- Flash patch and breakpoint (FPB) unit supports:
  - Two literal comparators
  - Six instruction comparators
- Data watchpoint and trace (DWT) unit
  - Four comparators
- Instrumentation trace macrocell (ITM)
- Embedded trace macrocell (ETM)
- Trace port interface unit (TPIU)
  - 4-bit parallel trace of ITM and ETM trace data
  - Serial wire output (SWO) trace of ITM data

4.8.1 DAP - Debug access port

An external debugger can access the device via the DAP.

The debug access port (DAP) implements a standard ARM® CoreSight™ serial wire debug port (SW-DP), which again implements the serial wire debug protocol (SWD). SWD is a two-pin serial interface, see SWDCLK and SWDIO in #unique_105/unique_105_Connect_42_fig.

In addition to the default access port in CPU (AHB-AP), the DAP includes a custom control access port (CTRL-AP). The CTRL-AP is described in more detail in CTRL-AP - Control access port on page 50.

Note:

- The SWDIO line has an internal pull-up resistor.
- The SWDCLK line has an internal pull-down resistor.

4.8.2 CTRL-AP - Control access port

The control access port (CTRL-AP) is a custom access port that enables control of the device when other access ports in the DAP are disabled by the access port protection.

Access port protection blocks the debugger from read and write access to all CPU registers and memory-mapped addresses. See the UICR register APPROTECT on page 44 for more information on enabling access port protection.

Control access port has the following features:

- Soft reset, see Reset on page 68 for more information
- Disabling of access port protection, which is the reason why CTRL-AP allows control of the device even when all other access ports in the DAP are disabled by the access port protection

Access port protection is disabled by issuing an ERASEALL command via CTRL-AP. This command will erase the flash, UICR, and RAM.

4.8.2.1 Registers

Register	Offset	Description
RESET	0x000	Soft reset triggered through CTRL-AP
ERASEALL	0x004	Erase all
ERASEALLSTATUS	0x008	Status register for the ERASEALL operation
APPROTECTSTATUS	0x00C	Status register for access port protection
IDR	0x0FC	CTRL-AP identification register, IDR

Table 12: Register overview

4.8.2.1.1 RESET

Address offset: 0x000

Soft reset triggered through CTRL-AP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RESET			Soft reset triggered through CTRL-AP. See Reset behavior in POWER chapter for more details.
			NoReset	0	Reset is not active
			Reset	1	Reset is active. Device is held in reset.

4.8.2.1.2 ERASEALL

Address offset: 0x004

Erase all

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	ERASEALL			Erase all flash and RAM
			NoOperation	0	No operation
			Erase	1	Erase all flash and RAM

4.8.2.1.3 ERASEALLSTATUS

Address offset: 0x008

Status register for the ERASEALL operation

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ERASEALLSTATUS			Status register for the ERASEALL operation
			Ready	0	ERASEALL is ready
			Busy	1	ERASEALL is busy (on-going)

4.8.2.1.4 APPROTECTSTATUS

Address offset: 0x00C

Status register for access port protection

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	APPROTECTSTATUS			Status register for access port protection
			Enabled	0	Access port protection enabled
			Disabled	1	Access port protection not enabled

4.8.2.1.5 IDR

Address offset: 0x0FC

CTRL-AP identification register, IDR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E E E D D D D C C C C C C B B B B A A A A A A A A
Reset 0x02880000				**0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	APID			AP identification
B	R	CLASS			Access port (AP) class
			NotDefined	0x0	No defined class
			MEMAP	0x8	Memory access port
C	R	JEP106ID			JEDEC JEP106 identity code
D	R	JEP106CONT			JEDEC JEP106 continuation code
E	R	REVISION			Revision

4.8.2.2 Electrical specification

4.8.2.2.1 Control access port

Symbol	Description	Min.	Typ.	Max.	Units
R_{pull}	Internal SWDIO and SWDCLK pull up/down resistance		13		kΩ
f_{SWDCLK}	SWDCLK frequency	0.125		8	MHz

4.8.3 Debug interface mode

Before an external debugger can access either CPU's access port (AHB-AP) or the control access port (CTRL-AP), the debugger must first request the device to power up via CxxxPWRUPREQ in the SWJ-DP.

If the device is in System OFF when power is requested via CxxxPWRUPREQ, the system will wake up and the DIF flag in RESETREAS on page 72 will be set. The device is in the debug interface mode as long as the debugger is requesting power via CxxxPWRUPREQ. Once the debugger stops requesting power via CxxxPWRUPREQ, the device is back in normal mode. Some peripherals behave differently in debug interface mode compared to normal mode. These differences are described in more detail in the chapters of the peripherals that are affected.

When a debug session is over, the external debugger must make sure to put the device back into normal mode since the overall power consumption is higher in debug interface mode than in normal mode.

For details on how to use the debug capabilities, read the debug documentation of your IDE.

4.8.4 Real-time debug

The nRF52840 supports real-time debugging.

Real-time debugging allows interrupts to execute to completion in real time when breakpoints are set in thread mode or lower priority interrupts. This enables developers to set breakpoints and single-step through the code without the risk of real-time event-driven threads running at higher priority failing. For example, this enables the device to continue to service the high-priority interrupts of an external controller or sensor without failure or loss of state synchronization while the developer steps through code in a low-priority thread.

4.8.5 Trace

The device supports ETM and ITM trace.

Trace data from the ETM and the ITM is sent to an external debugger via a 4-bit wide parallel trace port interface unit (TPIU), see TRACEDATA[0] through TRACEDATA[3] and TRACECLK in #unique_105/unique_105_Connect_42_fig.

In addition to parallel trace, the TPIU supports serial trace via the serial wire output (SWO) trace protocol. Parallel and serial trace cannot be used at the same time. ETM trace is only supported in parallel trace mode, while ITM trace is supported in both parallel and serial trace modes.

For details on how to use the trace capabilities, read the debug documentation of your IDE.

TPIU's trace pins are multiplexed with GPIOs, and SWO and TRACEDATA[0] use the same GPIO, see Pin assignments on page 524 for more information.

Trace speed is configured in the TRACECONFIG on page 89 register. The speed of the trace pins depends on the DRIVE setting of the GPIOs that the trace pins are multiplexed with. Only S0S1 and H0H1 drives are suitable for debugging. S0S1 is the default DRIVE at reset. If parallel or serial trace port signals are not fast enough in the debugging conditions, all GPIOs in use for tracing should be set to high drive (H0H1). The user shall make sure that DRIVE setting for these GPIOs is not overwritten by software during the debugging session.

4.8.5.1 Electrical specification

4.8.5.1.1 Trace port

Symbol	Description	Min.	Typ.	Max.	Units
T_{cyc}	Clock period, as defined by ARM (See Embedded Trace Macrocell Architecture Specification->Trace Port Physical Interface->Timing specifications on ARM Information Center)	62.5		500	ns

5 Power and clock management

5.1 Power management unit (PMU)

Power and clock management in nRF52840 is designed to automatically ensure maximum power efficiency.

The core of the power and clock management system is the power management unit (PMU) illustrated in Power management unit on page 54.

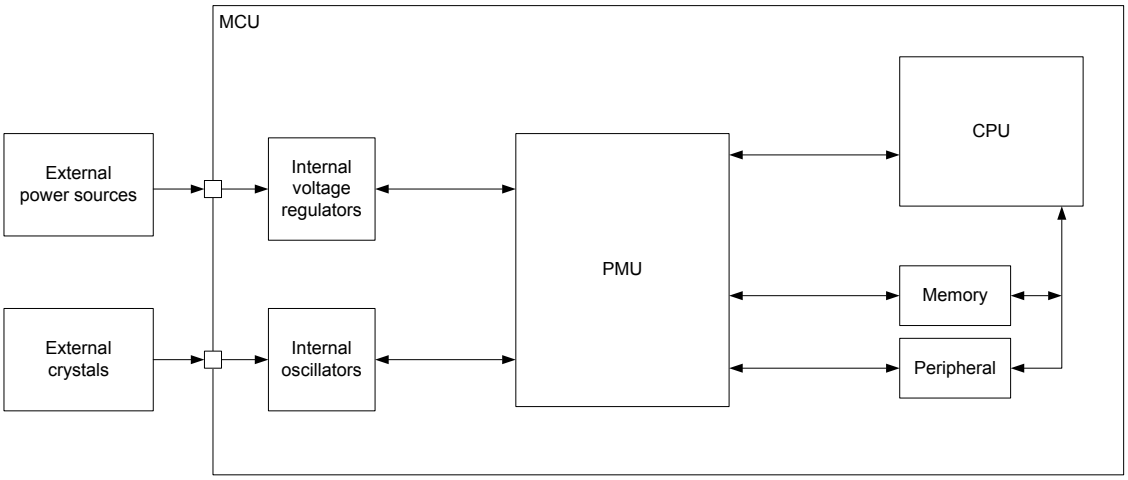

Figure 8: Power management unit

The PMU automatically detects which power and clock resources are required by the different components in the system at any given time. It will then start/stop and choose operation modes in supply regulators and clock sources, without user interaction, to achieve the lowest power consumption possible.

5.2 Current consumption

As the system is being constantly tuned by the Power management unit (PMU) on page 54, estimating the current consumption of an application can be challenging if the designer is not able to perform measurements directly on the hardware. To fascilitate the estimation process, a set of current consumption scenarios are provided to show the typical current drawn from the VDD supply.

Each scenario specifies a set of operations and conditions applying to the given scenario. Current consumption scenarios, common conditions on page 55 shows a set of common conditions used in all scenarios, unless otherwise is stated in the description of a given scenario. All scenarios are listed in Electrical specification on page 55

Condition	Value
Supply	3 V on VDD/VDDH (Normal voltage mode)
Temperature	25°C
CPU	WFI (wait for interrupt)/WFE (wait for event) sleep
Peripherals	All idle
Clock	Not running
Regulator	LDO
RAM	Full 256 kB retention
Compiler[2]	GCC v4.9.3 20150529 (arm-none-eabi-gcc). Compiler flags: -O0 -falign-functions=16 -fno-strict-aliasing -mcpu=cortex-m4 -mfloat-abi=soft -msoft-float -mthumb.
Cache enabled[2]	Yes
32 MHz crystal[3]	SMD 2520, 32 MHz, 10 pF +/- 10 ppm

Table 13: Current consumption scenarios, common conditions

5.2.1 Electrical specification

5.2.1.1 Sleep

Symbol	Description	Min.	Typ.	Max.	Units
$I_{ON\_RAMOFF\_EVENT}$	System ON, no RAM retention, wake on any event		0.97		µA
$I_{ON\_RAMON\_EVENT}$	System ON, full 256 kB RAM retention, wake on any event		2.35		µA
$I_{ON\_RAMON\_POF}$	System ON, full 256 kB RAM retention, wake on any event, power-fail comparator enabled		2.35		µA
$I_{ON\_RAMON\_GPIOTE}$	System ON, full 256 kB RAM retention, wake on GPIOTE input (event mode)		17.37		µA
$I_{ON\_RAMON\_GPIOTEPORT}$	System ON, full 256 kB RAM retention, wake on GPIOTE PORT event		2.36		µA
$I_{ON\_RAMOFF\_RTC}$	System ON, no RAM retention, wake on RTC (running from LFRC clock)		1.5		µA
$I_{ON\_RAMON\_RTC}$	System ON, full 256 kB RAM retention, wake on RTC (running from LFRC clock)		3.16		µA
$I_{OFF\_RAMOFF\_RESET}$	System OFF, no RAM retention, wake on reset		0.40		µA
$I_{OFF\_RAMOFF\_LPCOMP}$	System OFF, no RAM retention, wake on LPCOMP		0.86		µA
$I_{OFF\_RAMON\_RESET}$	System OFF, full 256 kB RAM retention, wake on reset		1.86		µA
$I_{ON\_RAMOFF\_EVENT\_5V}$	System ON, no RAM retention, wake on any event, 5 V supply on VDDH, REG0 output = 3.3 V		1.29		µA
$I_{OFF\_RAMOFF\_RESET\_5V}$	System OFF, no RAM retention, wake on reset, 5 V supply on VDDH, REG0 output = 3.3 V		0.95		µA

[2] Applying only when CPU is running from flash memory
[3] Applying only when HFXO is running

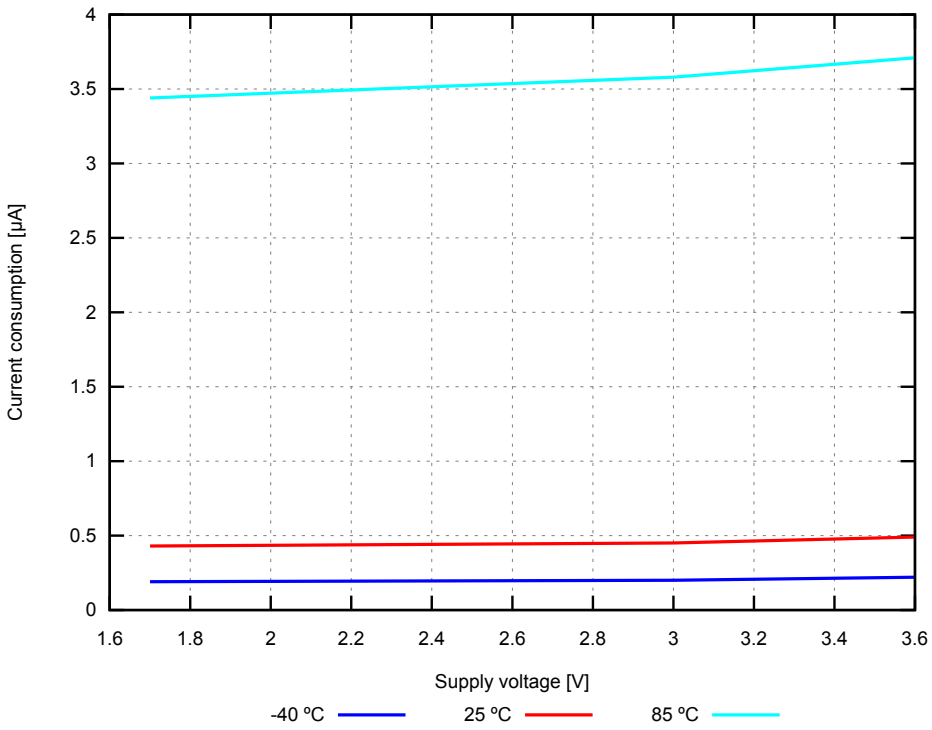

Figure 9: System OFF, no RAM retention, wake on reset (typical values)

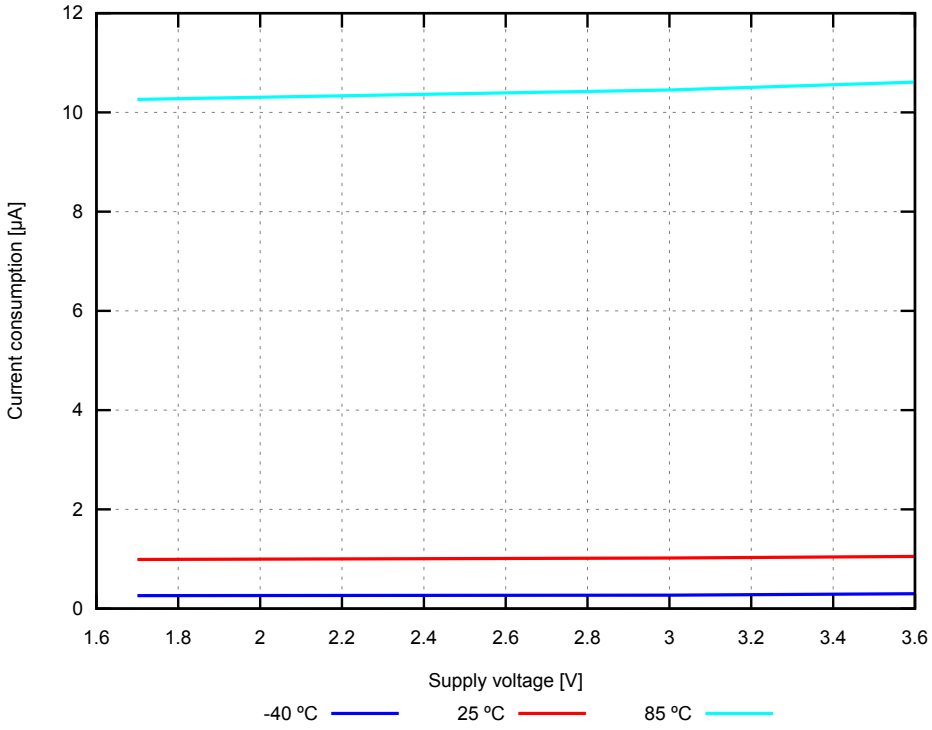

Figure 10: System ON, no RAM retention, wake on any event (typical values)

5.2.1.2 COMP active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{COMP,LP}$	COMP enabled, low power mode		30.1		µA
$I_{COMP,NORM}$	COMP enabled, normal mode		31.8		µA
$I_{COMP,HS}$	COMP enabled, high-speed mode		35.1		µA

5.2.1.3 CPU running

Symbol	Description	Min.	Typ.	Max.	Units
I_{CPU0}	CPU running CoreMark @64 MHz from flash, Clock = HFXO, Regulator = DC/DC		3.3		mA
I_{CPU1}	CPU running CoreMark @64 MHz from flash, Clock = HFXO		6.3		mA
I_{CPU2}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO, Regulator = DC/DC		2.8		mA
I_{CPU3}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO		5.2		mA
I_{CPU4}	CPU running CoreMark @64 MHz from flash, Clock = HFINT, Regulator = DC/DC		3.1		mA

5.2.1.4 NFCT active

Symbol	Description	Min.	Typ.	Max.	Units
I_{sense}	Current in SENSE STATE[4]		100		nA
$I_{activated}$	Current in ACTIVATED STATE		400		µA

5.2.1.5 Radio transmitting/receiving

Symbol	Description	Min.	Typ.	Max.	Units
I_{RADIO_TX0}	Radio transmitting @ 8 dBm output power, 1 Mbps *Bluetooth*® low energy (BLE) mode, Clock = HFXO, Regulator = DC/DC		16.4		mA
I_{RADIO_TX1}	Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		6.40		mA
I_{RADIO_TX2}	Radio transmitting @ -40 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		3.83		mA
I_{RADIO_TX3}	Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO		10.8		mA
I_{RADIO_TX4}	Radio transmitting @ -40 dBm output power, 1 Mbps BLE mode, Clock = HFXO		4.82		mA
I_{RADIO_TX5}	Radio transmitting @ 0 dBm output power, 250 kbit/s IEE 802.15.4-2006 mode, Clock = HFXO, Regulator = DC/DC		6.40		mA
I_{RADIO_RX0}	Radio receiving @ 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		6.26		mA
I_{RADIO_RX1}	Radio receiving @ 1 Mbps BLE mode, Clock = HFXO		10.1		mA
I_{RADIO_RX2}	Radio receiving @ 250 kbit/s IEE 802.15.4-2006 mode, Clock = HFXO, Regulator = DC/DC		6.53		mA

[4] This current does not apply when in NFC field

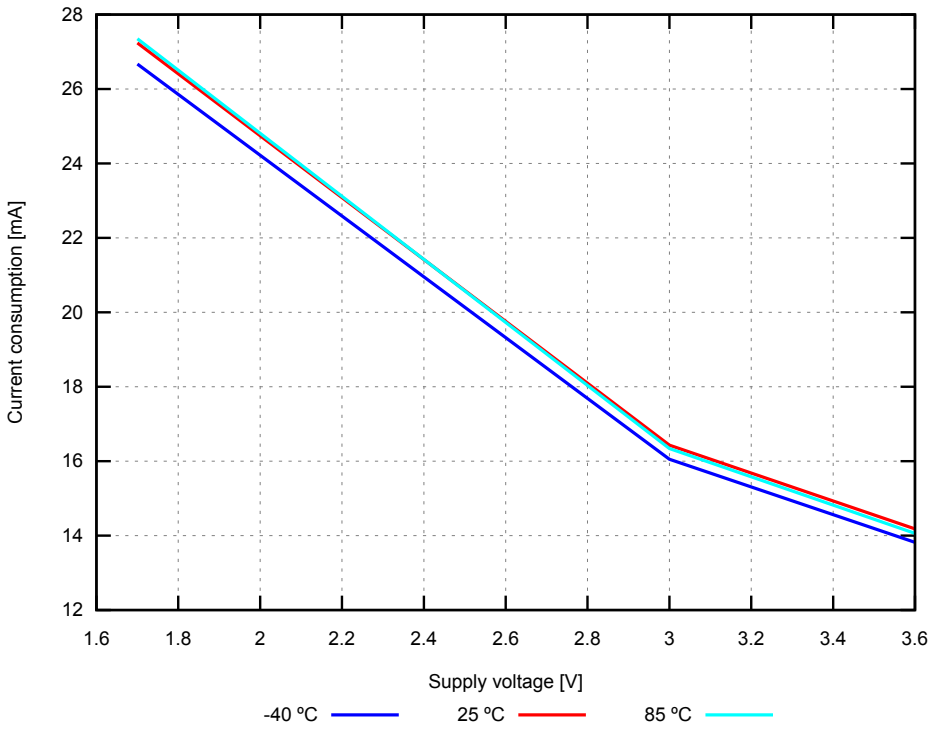

Figure 11: Radio transmitting @ 8 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC (typical values)

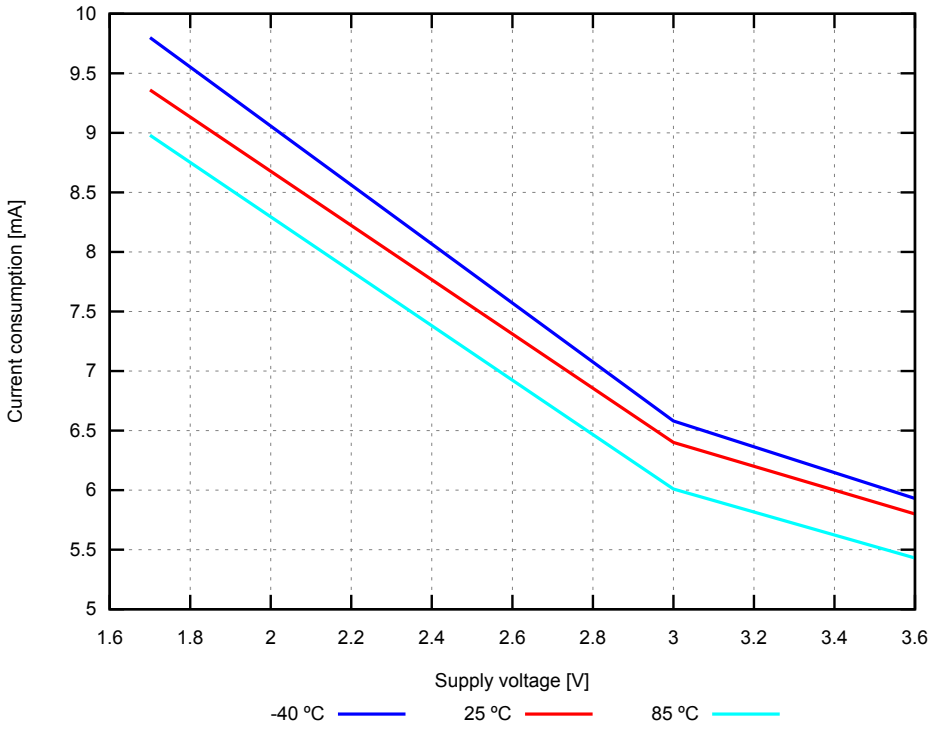

Figure 12: Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC (typical values)

5.2.1.6 RNG active

Symbol	Description	Min.	Typ.	Max.	Units
I_{RNG0}	RNG running		635		µA

5.2.1.7 SAADC active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{SAADC,RUN}$	SAADC sampling @ 16 ksps, Acquisition time = 20 µs, Clock = HFXO, Regulator = DC/DC		1.24		mA

5.2.1.8 TEMP active

Symbol	Description	Min.	Typ.	Max.	Units
I_{TEMP0}	TEMP started		1.05		mA

5.2.1.9 TIMER running

Symbol	Description	Min.	Typ.	Max.	Units
I_{TIMER0}	One TIMER instance running @ 1 MHz, Clock = HFINT		418		µA
I_{TIMER1}	Two TIMER instances running @ 1 MHz, Clock = HFINT		418		µA
I_{TIMER2}	One TIMER instance running @ 1 MHz, Clock = HFXO		646		µA
I_{TIMER3}	One TIMER instance running @ 16 MHz, Clock = HFINT		595		µA
I_{TIMER4}	One TIMER instance running @ 16 MHz, Clock = HFXO		823		µA

5.2.1.10 WDT active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{WDT,STARTED}$	WDT started		3.1		µA

5.2.1.11 Compounded

Symbol	Description	Min.	Typ.	Max.	Units
I_{S0}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps *Bluetooth*® low energy (BLE) mode, Clock = HFXO, Regulator = DC/DC		8.1		mA
I_{S1}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		8.6		mA
I_{S2}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO		15.4		mA
I_{S3}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps BLE mode, Clock = HFXO		16.2		mA
I_{S4}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC, 5 V supply on VDDH, REG0 output = 3.3 V		11.9		mA
I_{S5}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC, 5 V supply on VDDH, REG0 output = 3.3 V		12.7		mA

5.3 POWER — Power supply

The power supply consists of a number of LDO and DC/DC regulators that are utilized to maximize the system's power efficiency.

This device has the following power supply features:

- On-chip LDO and DC/DC regulators
- Global System ON/OFF modes
- Individual RAM section power control for all system modes
- Analog or digital pin wakeup from System OFF
- Supervisor hardware to manage power-on reset, brownout, and power failure
- Auto-controlled refresh modes for LDO and DC/DC regulators to maximize efficiency
- External circuitry supply
- Separate USB supply

5.3.1 Main supply

The main supply voltage is connected to the VDD/VDDH pins. The system will enter one of two supply voltage modes, normal or high voltage mode, depending on how the supply voltage is connected to these pins.

Normal voltage mode is entered when the supply voltage is connected to both the VDD and VDDH pins (so that VDD equals VDDH).

High voltage mode is entered when the supply voltage is only connected to the VDDH pin and the VDD pin is not connected to any voltage supply.

The register MAINREGSTATUS on page 76 can be used for reading out the current supply voltage mode.

For the supply voltage range of the two supply voltage modes, see Regulator operating conditions on page 78.

5.3.1.1 Main voltage regulators

The system contains two main supply regulator stages, REG0 and REG1.

Each regulator stage has the following regulator type options:

- Low-dropout regulator (LDO)
- Buck regulator (DC/DC)

In normal voltage mode, only the REG1 regulator stage is used and the REG0 stage is automatically disabled. In high voltage mode, both regulator stages (REG0 and REG1) are used. The output voltage of REG0 can be configured in register REGOUT0 on page 45. This output voltage is connected to VDD and is the input voltage to REG1.

By default, the LDO regulators are enabled and the DC/DC regulators are disabled. Registers DCDCEN0 on page 76 and DCDCEN on page 76 are used to independently enable the DC/DC regulators for the two stages (REG0 and REG1 respectively).

When a DC/DC converter is enabled, the LDO for the corresponding regulator stage will be disabled. External LC filters must be connected for each of the DC/DC regulators being used. The advantage of using a DC/DC regulator is that the overall power consumption is normally reduced as the efficiency of such a regulator is higher than that of a LDO. The efficiency benefit of using a DC/DC regulator becomes particularly prominent when the regulator voltage drop (difference between input and output voltage) is high. The efficiency of internal regulators vary with the supply voltage and the current drawn from the regulators.

Note: Do not enable DC/DC regulator without an external LC filter being connected as this will inhibit device operation, including debug access, until an LC filter is connected.

5.3.1.2 GPIO levels

The GPIO high reference voltage always equals the level on the VDD pin.

In normal voltage mode, the GPIO high level equals the voltage supplied to the VDD pin, and in high voltage mode it equals the level specified in register REGOUT0 on page 45.

5.3.1.3 External circuitry supply

In high voltage mode, the output from REG0 can be used to supply external circuitry from the VDD pin.

The VDD output voltage is configured in the register REGOUT0 on page 45.

The supported output voltage range depends on the supply voltage provided on the VDDH pin. Minimum difference between voltage supplied on the VDDH pin and the voltage output on the VDD pin is defined by the $V_{REG0,DROP}$ parameter in Regulator specifications, REG0 stage on page 78.

Supplying external circuitry is allowed in both System OFF and System ON mode.

Note: The maximum allowed current drawn by external circuitry is dependent on the total internal current draw. The maximum current that can be drawn externally from REG0 is defined in Regulator specifications, REG0 stage on page 78).

5.3.1.4 Regulator configuration examples

The voltage regulators can be configured in several ways, depending on the selected supply voltage mode (normal/high) and the regulator type option (LDO or DC/DC).

Four configuration examples are illustrated in images below.

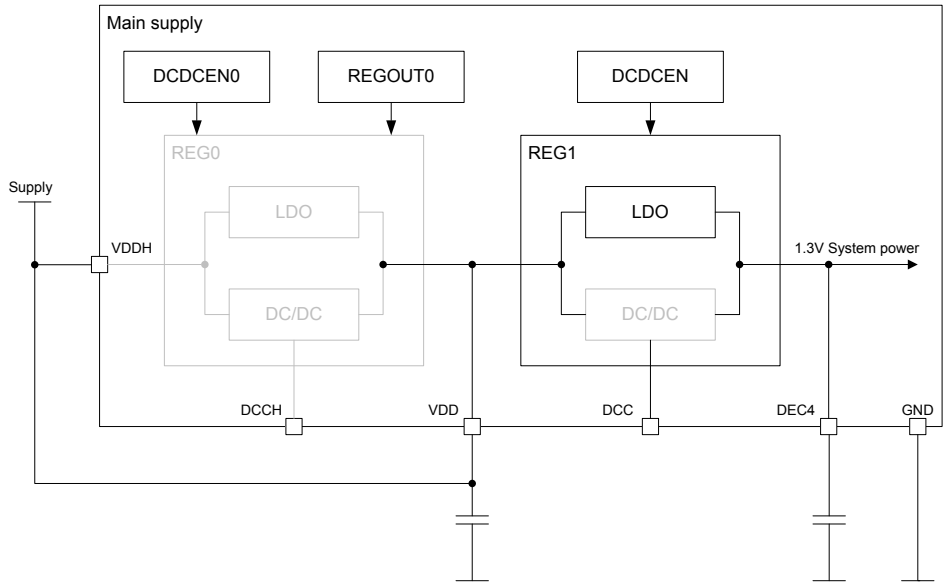

Figure 13: Normal voltage mode, LDO only

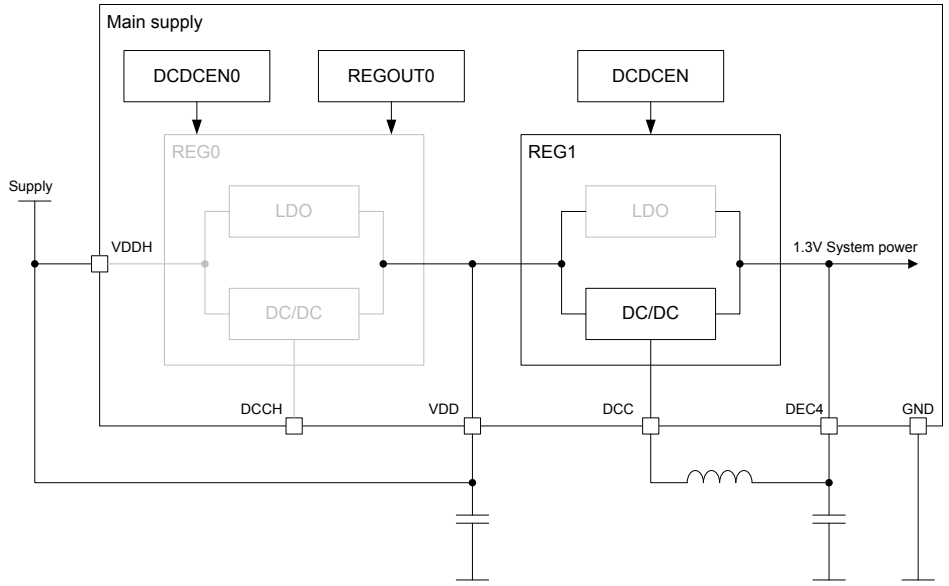

Figure 14: Normal voltage mode, DC/DC REG1 enabled

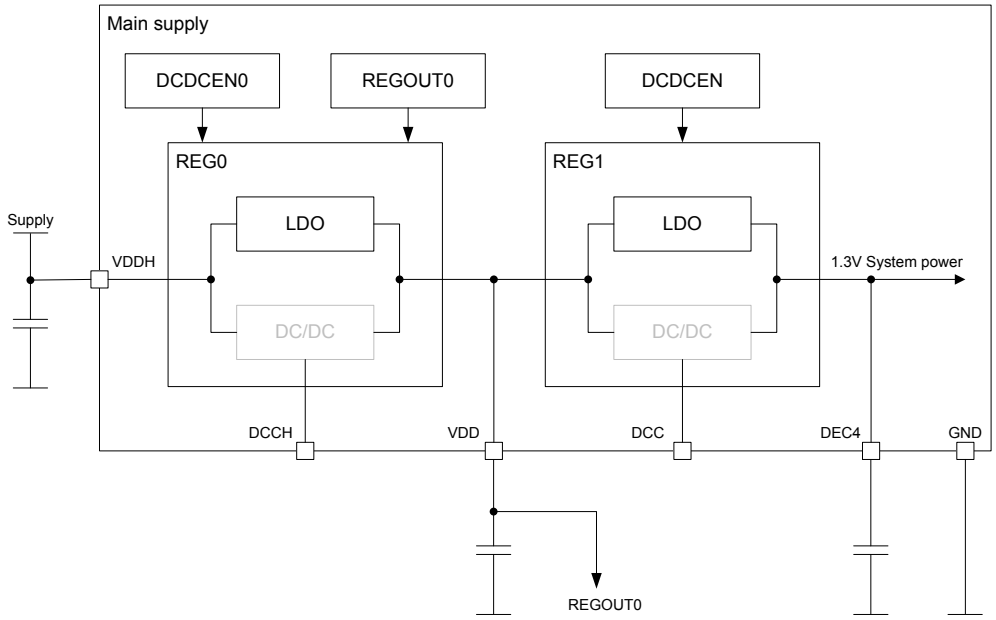

Figure 15: High voltage mode, LDO only

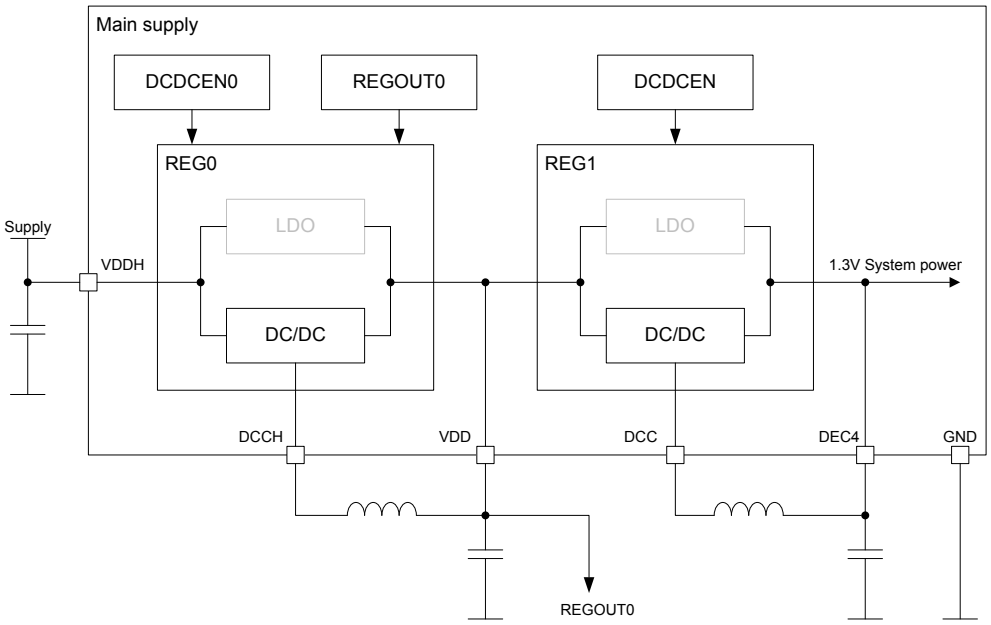

Figure 16: High voltage mode, DC/DC for REG0 and REG1 enabled

5.3.1.5 Power supply supervisor

The power supply supervisor enables monitoring of the connected power supply.

The power supply supervisor provides:

- Power-on reset, signalling to the circuit when a supply is connected.
- An optional power-fail comparator (POF), to signal the application when the supply voltages drop below a configured threshold.
- A fixed brownout reset detector, to hold the system in reset when the voltage is too low for safe operation.

The power supply supervisor is illustrated in Power supply supervisor on page 64. To enable and configure the power-fail comparator, see the register POFCON on page 74.

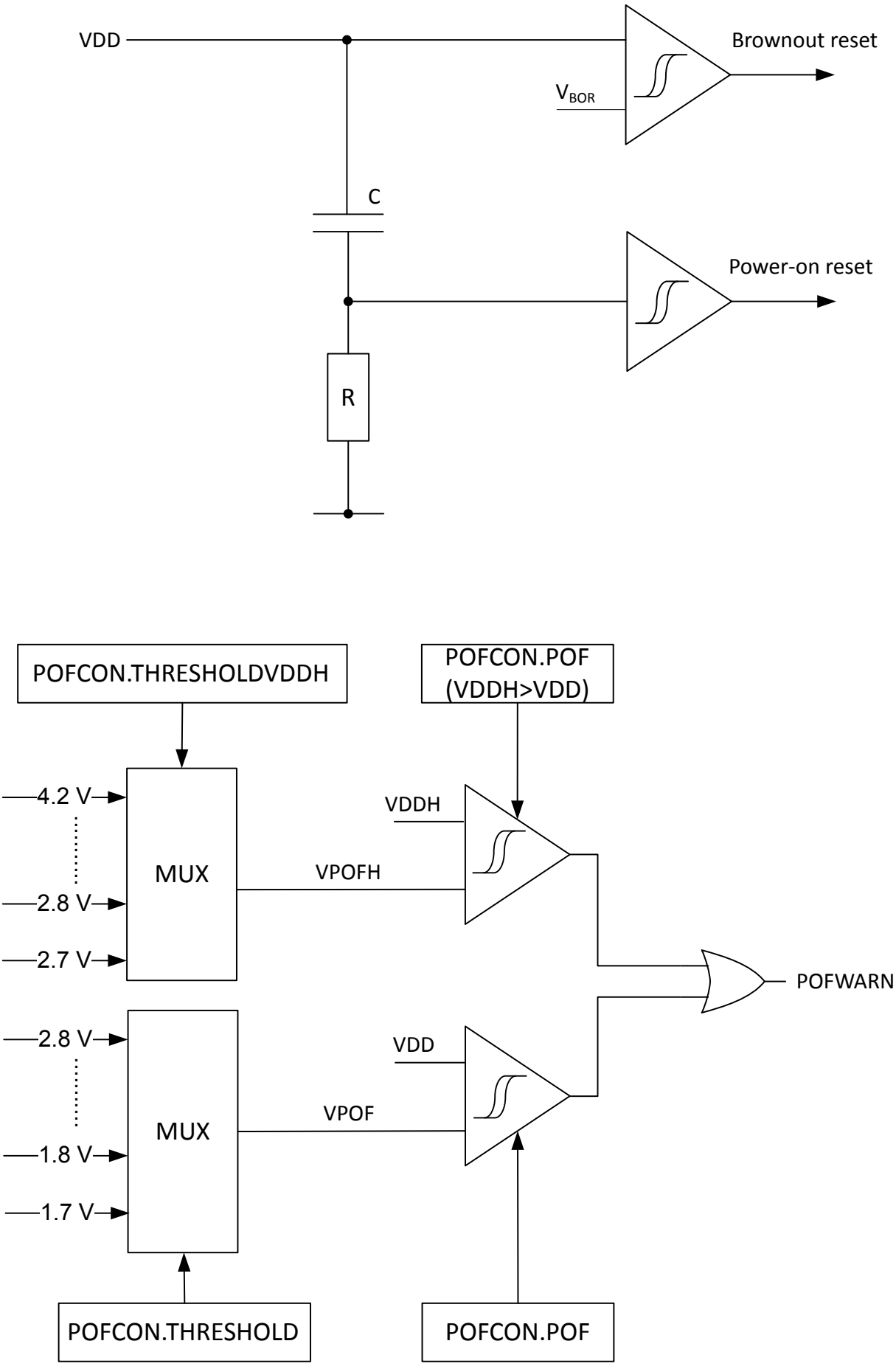

Figure 17: Power supply supervisor

5.3.1.6 Power-fail comparator

Using the power-fail comparator (POF) is optional. When enabled, it can provide the CPU an early warning of an impending power supply failure.

To enable and configure the power-fail comparator, see the register POFCON on page 74.

When the supply voltage falls below the defined threshold, the power-fail comparator will generate an event (POFWARN) which can be used by an application to prepare for power failure. This event will also be generated if the supply voltage is already below the threshold at the time the power-fail comparator is enabled, or if the threshold is re-configured to a level above the supply voltage.

If the power failure warning is enabled and the supply voltage is below the threshold, the power-fail comparator will prevent the NVMC from performing write operations to the flash.

The comparator features a hysteresis of V_{HYST}, as illustrated in Power-fail comparator (BOR = brownout reset) on page 65.

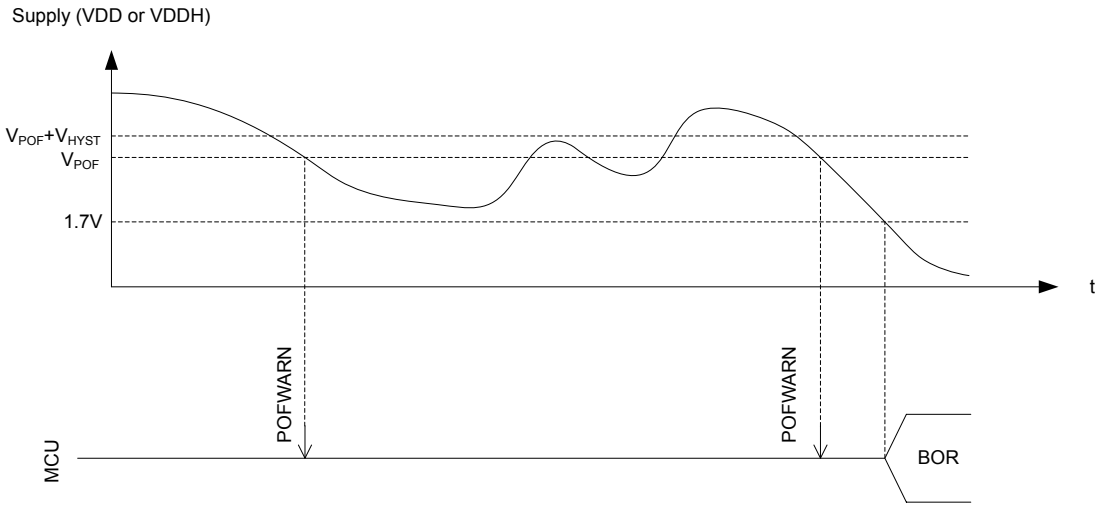

Figure 18: Power-fail comparator (BOR = brownout reset)

To save power, the power-fail comparator is not active in System OFF or in System ON when HFCLK is not running.

5.3.2 USB supply

When using the USB peripheral, a 5 V USB supply needs to be provided on the VBUS pin.

The USB peripheral has a dedicated internal voltage regulator for converting the VBUS supply to 3.3 V used by the USB signalling interface (D+ and D- lines, and pull-up on D+). The rest of the USB peripheral (USBD) is supplied through the main supply like any other on-chip feature. As a consequence, both VBUS and either VDDH or VDD supplies are required for USB peripheral operation.

When VBUS rises into its valid range, the software is notified through a USBDETECTED event. A USBREMOVED event is sent when VBUS goes below its valid range. Use these events to implement the USBD start-up sequence described in the USBD chapter.

When VBUS rises into its valid range while the device is in System OFF, the device resets and transitions to System ON mode. The RESETREAS register will have the VBUS bit set to indicate the source of the wake-up.

See VBUS detection specifications on page 79 for the levels at which the events are sent ($V_{BUS,DETECT}$ and $V_{BUS,REMOVE}$) or at which the system is woken up from System OFF ($V_{BUS,DETECT}$).

When the USBD peripheral is enabled through the ENABLE register and VBUS is detected, the regulator is turned on. A USBPWRRDY event is sent when the regulator's worst case settling time has elapsed, indicating to the software that it can enable the USB pull-up to signal a USB connection to the host.

The software can read the state of the VBUS detection and regulator output readiness at any time through the USBREGSTATUS register.

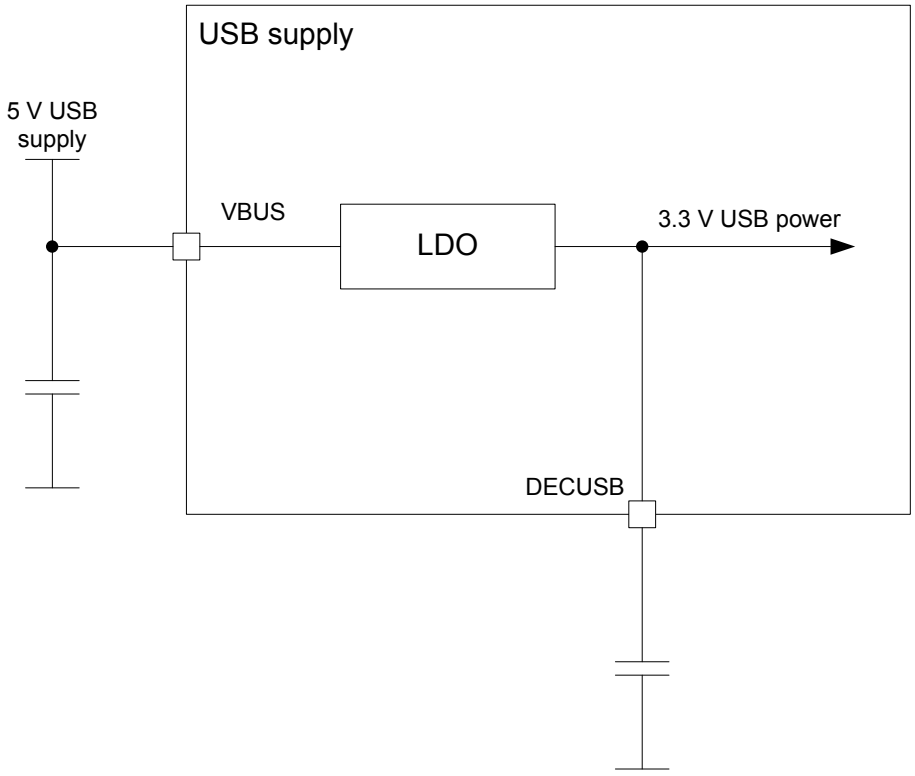

Figure 19: USB voltage regulator

To ensure stability, the input and output of the USB regulator need to be decoupled with a suitable decoupling capacitor. See Reference circuitry on page 528 for the recommended values.

5.3.3 System OFF mode

System OFF is the deepest power saving mode the system can enter. In this mode, the system's core functionality is powered down and all ongoing tasks are terminated.

The device can be put into System OFF mode using the POWER register interface. When in System OFF mode, the device can be woken up through one of the following signals:

1. The DETECT signal, optionally generated by the GPIO peripheral.
2. The ANADETECT signal, optionally generated by the LPCOMP module.
3. The SENSE signal, optionally generated by the NFC module to wake-on-field.
4. Detecting a valid USB voltage on the VBUS pin ($V_{BUS,DETECT}$).
5. A reset.

The system is reset when it wakes up from the System OFF mode.

One or more RAM sections can be retained in System OFF mode, depending on the settings in the RAM[n].POWER registers. RAM[n].POWER are retained registers. Note that these registers are usually overwritten by the start-up code provided with the nRF application examples.

Before entering the System OFF mode, the user must make sure that all on-going EasyDMA transactions have been completed. See peripheral specific chapters for more information about how to acquire the status of EasyDMA transactions.

5.3.3.1 Emulated System OFF mode

If the device is in debug interface mode, System OFF will be emulated to secure that all required resources needed for debugging are available during System OFF.

See Debug and trace on page 49 for more information. Required resources needed for debugging include the following key components: Debug and trace on page 49, CLOCK — Clock control on page 80, POWER — Power supply on page 60, NVMC — Non-volatile memory controller on page 23, CPU on page 18, flash, and RAM. Since the CPU is kept on in an emulated System OFF mode, it is recommended to add an infinite loop directly after entering System OFF, to prevent the CPU from executing code that normally should not be executed.

5.3.4 System ON mode

System ON is the default state after power-on reset. In System ON, all functional blocks such as the CPU or peripherals, can be in IDLE or RUN mode, depending on the configuration set by the software and the state of the application executing.

Register RESETREAS on page 72 provides information about the source causing the wakeup or reset.

The system can switch the appropriate internal power sources on and off, depending on how much power is needed at any given time. The power requirement of a peripheral is directly related to its activity level, and the activity level of a peripheral is usually raised and lowered when specific tasks are triggered or events are generated.

5.3.4.1 Sub power modes

In System ON mode, when both the CPU and all the peripherals are in IDLE mode, the system can reside in one of the two sub power modes.

The sub power modes are:

- Constant latency
- Low power

In constant latency mode, the CPU wakeup latency and the PPI task response are constant and kept at a minimum. This is secured by forcing a set of basic resources to be turned on while in sleep. Having a constant and predictable latency is at cost of having increased power consumption. The constant latency mode is selected by triggering the CONSTLAT task.

In low power mode, the automatic power management system described in System ON mode on page 67 ensures that the most efficient supply option is chosen to save most power. Having the lowest power possible is at cost of having a varying CPU wakeup latency and PPI task response. The low power mode is selected by triggering the LOWPWR task.

When the system enters System ON mode, it is by default in low power sub power mode.

5.3.5 RAM power control

The RAM power control registers are used for configuring the following:

- What RAM sections to be retained during System OFF
- What RAM sections to be retained and accessible during System ON

In System OFF, retention of a RAM section is configured in the RETENTION field of the corresponding register RAM[n].POWER (n=0..8) on page 76.

In System ON, retention and accessibility for a RAM section is configured in the RETENTION and POWER fields of the corresponding register RAM[n].POWER (n=0..8) on page 76.

The table below summaries the behavior of these registers.

Configuration			RAM section status	
System on/off	RAM[n].POWER.POWER	RAM[n].POWER.RETENTION	Accessible	Retained
Off	x	Off	No	No
Off	x	On	No	Yes
On	Off	Off	No	No
On	Off[1]	On	No	Yes
On	On	x	Yes	Yes

Table 14: RAM section configuration. x = don't care.

The advantage of not retaining RAM contents is that the overall current consumption is reduced.

See chapter Memory on page 19 for more information on RAM sections.

5.3.6 Reset

Several sources may trigger a reset.

After a reset has occurred, register RESETREAS can be read to determine which source has triggered the reset.

5.3.6.1 Power-on reset

The power-on reset generator initializes the system at power-on.

The system is held in a reset state until the power supply has reached the minimum operating voltage and the internal voltage regulators have started.

5.3.6.2 Pin reset

A pin reset is generated when the physical reset pin on the device is asserted.

Pin reset is configured via the PSELRESET[0] and PSELRESET[1] registers.

5.3.6.3 Wakeup from System OFF mode reset

The device is reset when it wakes up from System OFF mode.

The debug access port (DAP) is not reset following a wake up from System OFF mode if the device is in debug interface mode. See chapter Debug and trace on page 49 for more information.

5.3.6.4 Soft reset

A soft reset is generated when the SYSRESETREQ bit of the application interrupt and reset control register (AIRCR) in the ARM® core is set.

See ARM documentation for more details.

A soft reset can also be generated via the register RESET on page 50 in the CTRL-AP.

5.3.6.5 Watchdog reset

A watchdog reset is generated when the watchdog times out.

See chapter WDT — Watchdog timer on page 519 for more information.

5.3.6.6 Brownout reset

The brownout reset generator puts the system in reset state if VDD drops below the brownout reset (BOR) threshold.

[1] Not useful setting. RAM section power off gives negligible reduction in current consumption when retention is on.

See section Power fail comparator on page 79 for more information.

5.3.6.7 Retained registers

A retained register is a register that will retain its value in System OFF mode and through a reset, depending on reset source. See the individual peripheral chapters for information on which of their registers are retained.

5.3.6.8 Reset behavior

What targets are reset by the various reset sources are summarized in the table below.

Reset source	Reset target								
	CPU	Peripherals	GPIO	Debug[a]	SWJ-DP	RAM	WDT	Retained registers	RESETREAS
CPU lockup [6]	x	x	x						
Soft reset	x	x	x						
Wakeup from System OFF mode reset	x	x		x [7]		x [8]			
Watchdog reset [9]	x	x	x	x		x	x	x	
Pin reset	x	x	x	x		x	x	x	
Brownout reset	x	x	x	x	x	x	x	x	x
Power-on reset	x	x	x	x	x	x	x	x	x

Note: The RAM is never reset, but depending on a reset source the content of RAM may be corrupted.

5.3.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	POWER	POWER	Power control	

Table 15: Instances

Register	Offset	Description
TASKS_CONSTLAT	0x78	Enable constant latency mode
TASKS_LOWPWR	0x7C	Enable low power mode (variable latency)
EVENTS_POFWARN	0x108	Power failure warning
EVENTS_SLEEPENTER	0x114	CPU entered WFI/WFE sleep
EVENTS_SLEEPEXIT	0x118	CPU exited WFI/WFE sleep
EVENTS_USBDETECTED	0x11C	Voltage supply detected on VBUS
EVENTS_USBREMOVED	0x120	Voltage supply removed from VBUS
EVENTS_USBPWRRDY	0x124	USB 3.3 V supply ready
INTENSET	0x304	Enable interrupt

5 All debug components excluding SWJ-DP. See Debug and trace on page 49 chapter for more information about the different debug components.

6 Reset from CPU lockup is disabled if the device is in debug interface mode. CPU lockup is not possible in System OFF.

7 The debug components will not be reset if the device is in debug interface mode.

8 RAM is not reset on wakeup from System OFF mode. Whole RAM or parts of it may not be retained after the device has entered System OFF mode, depending on the settings in the RAM registers.

9 Watchdog reset is not available in System OFF.

Register	Offset	Description	
INTENCLR	0x308	Disable interrupt	
RESETREAS	0x400	Reset reason	
RAMSTATUS	0x428	RAM status register	Deprecated
USBREGSTATUS	0x438	USB supply status	
SYSTEMOFF	0x500	System OFF register	
POFCON	0x510	Power-fail comparator configuration	
GPREGRET	0x51C	General purpose retention register	
GPREGRET2	0x520	General purpose retention register	
DCDCEN	0x578	Enable DC/DC converter for REG1 stage.	
DCDCEN0	0x580	Enable DC/DC converter for REG0 stage.	
MAINREGSTATUS	0x640	Main supply status	
RAM[0].POWER	0x900	RAM0 power control register	
RAM[0].POWERSET	0x904	RAM0 power control set register	
RAM[0].POWERCLR	0x908	RAM0 power control clear register	
RAM[1].POWER	0x910	RAM1 power control register	
RAM[1].POWERSET	0x914	RAM1 power control set register	
RAM[1].POWERCLR	0x918	RAM1 power control clear register	
RAM[2].POWER	0x920	RAM2 power control register	
RAM[2].POWERSET	0x924	RAM2 power control set register	
RAM[2].POWERCLR	0x928	RAM2 power control clear register	
RAM[3].POWER	0x930	RAM3 power control register	
RAM[3].POWERSET	0x934	RAM3 power control set register	
RAM[3].POWERCLR	0x938	RAM3 power control clear register	
RAM[4].POWER	0x940	RAM4 power control register	
RAM[4].POWERSET	0x944	RAM4 power control set register	
RAM[4].POWERCLR	0x948	RAM4 power control clear register	
RAM[5].POWER	0x950	RAM5 power control register	
RAM[5].POWERSET	0x954	RAM5 power control set register	
RAM[5].POWERCLR	0x958	RAM5 power control clear register	
RAM[6].POWER	0x960	RAM6 power control register	
RAM[6].POWERSET	0x964	RAM6 power control set register	
RAM[6].POWERCLR	0x968	RAM6 power control clear register	
RAM[7].POWER	0x970	RAM7 power control register	
RAM[7].POWERSET	0x974	RAM7 power control set register	
RAM[7].POWERCLR	0x978	RAM7 power control clear register	
RAM[8].POWER	0x980	RAM8 power control register	
RAM[8].POWERSET	0x984	RAM8 power control set register	
RAM[8].POWERCLR	0x988	RAM8 power control clear register	

Table 16: Register overview

5.3.7.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	POFWARN			Write '1' to enable interrupt for POFWARN event See EVENTS_POFWARN
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	SLEEPENTER			Write '1' to enable interrupt for SLEEPENTER event See EVENTS_SLEEPENTER
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	SLEEPEXIT			Write '1' to enable interrupt for SLEEPEXIT event See EVENTS_SLEEPEXIT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	USBDETECTED			Write '1' to enable interrupt for USBDETECTED event See EVENTS_USBDETECTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	USBREMOVED			Write '1' to enable interrupt for USBREMOVED event See EVENTS_USBREMOVED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	USBPWRRDY			Write '1' to enable interrupt for USBPWRRDY event See EVENTS_USBPWRRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.3.7.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	POFWARN			Write '1' to disable interrupt for POFWARN event See EVENTS_POFWARN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	SLEEPENTER			Write '1' to disable interrupt for SLEEPENTER event See EVENTS_SLEEPENTER
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	SLEEPEXIT			Write '1' to disable interrupt for SLEEPEXIT event See EVENTS_SLEEPEXIT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	USBDETECTED			Write '1' to disable interrupt for USBDETECTED event See EVENTS_USBDETECTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	USBREMOVED			Write '1' to disable interrupt for USBREMOVED event See EVENTS_USBREMOVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	USBPWRRDY			Write '1' to disable interrupt for USBPWRRDY event See EVENTS_USBPWRRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.3.7.3 RESETREAS

Address offset: 0x400

Reset reason

Unless cleared, the RESETREAS register will be cumulative. A field is cleared by writing '1' to it. If none of the reset sources are flagged, this indicates that the chip was reset from the on-chip reset generator, which will indicate a power-on-reset or a brownout reset.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RESETPIN			Reset from pin-reset detected
			NotDetected	0	Not detected
			Detected	1	Detected
B	RW	DOG			Reset from watchdog detected
			NotDetected	0	Not detected
			Detected	1	Detected
C	RW	SREQ			Reset from soft reset detected
			NotDetected	0	Not detected
			Detected	1	Detected
D	RW	LOCKUP			Reset from CPU lock-up detected
			NotDetected	0	Not detected
			Detected	1	Detected
E	RW	OFF			Reset due to wake up from System OFF mode when wakeup is triggered from DETECT signal from GPIO
			NotDetected	0	Not detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Detected	1	Detected
F	RW	LPCOMP			Reset due to wake up from System OFF mode when wakeup is triggered from ANADETECT signal from LPCOMP
			NotDetected	0	Not detected
			Detected	1	Detected
G	RW	DIF			Reset due to wake up from System OFF mode when wakeup is triggered from entering into debug interface mode
			NotDetected	0	Not detected
			Detected	1	Detected
H	RW	NFC			Reset due to wake up from System OFF mode by NFC field detect
			NotDetected	0	Not detected
			Detected	1	Detected
I	RW	VBUS			Reset due to wake up from System OFF mode by VBUS rising into valid range
			NotDetected	0	Not detected
			Detected	1	Detected

5.3.7.4 RAMSTATUS (Deprecated)

Address offset: 0x428

RAM status register

Since this register is deprecated the following substitutions have been made: RAM block 0 is equivalent to a block comprising RAM0.S0 and RAM1.S0, RAM block 1 is equivalent to a block comprising RAM2.S0 and RAM3.S0, RAM block 2 is equivalent to a block comprising RAM4.S0 and RAM5.S0 and RAM block 3 is equivalent to a block comprising RAM6.S0 and RAM7.S0. A RAM block field will indicate ON as long as any of the RAM sections associated with a block are on.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-D	R	RAMBLOCK[i] (i=0..3)			RAM block i is on or off/powering up
			Off	0	Off
			On	1	On

5.3.7.5 USBREGSTATUS

Address offset: 0x438

USB supply status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	VBUSDETECT			VBUS input detection status (USBDETECTED and USBREMOVED events are derived from this information)
			NoVbus	0	VBUS voltage below valid threshold
			VbusPresent	1	VBUS voltage above valid threshold
B	R	OUTPUTRDY			USB supply output settling time elapsed
			NotReady	0	USBREG output settling time not elapsed
			Ready	1	USBREG output settling time elapsed (same information as USBPWRRDY event)

5.3.7.6 SYSTEMOFF

Address offset: 0x500

System OFF register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	W	SYSTEMOFF			Enable System OFF mode
			Enter	1	Enable System OFF mode

5.3.7.7 POFCON

Address offset: 0x510

Power-fail comparator configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D B B B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	POF			Enable or disable power failure warning
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	THRESHOLD			Power-fail comparator threshold setting. This setting applies both for normal voltage mode (supply connected to both VDD and VDDH) and high voltage mode (supply connected to VDDH only). Values 0-3 set threshold below 1.7 V and should not be used as brown out detection will be activated before power failure warning on such low voltages.
			V17	4	Set threshold to 1.7 V
			V18	5	Set threshold to 1.8 V
			V19	6	Set threshold to 1.9 V
			V20	7	Set threshold to 2.0 V
			V21	8	Set threshold to 2.1 V
			V22	9	Set threshold to 2.2 V
			V23	10	Set threshold to 2.3 V
			V24	11	Set threshold to 2.4 V
			V25	12	Set threshold to 2.5 V
			V26	13	Set threshold to 2.6 V

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D B B B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			V27	14	Set threshold to 2.7 V
			V28	15	Set threshold to 2.8 V
D	RW	THRESHOLDVDDH			Power-fail comparator threshold setting for high voltage mode (supply connected to VDDH only). This setting does not apply for normal voltage mode (supply connected to both VDD and VDDH).
			V27	0	Set threshold to 2.7 V
			V28	1	Set threshold to 2.8 V
			V29	2	Set threshold to 2.9 V
			V30	3	Set threshold to 3.0 V
			V31	4	Set threshold to 3.1 V
			V32	5	Set threshold to 3.2 V
			V33	6	Set threshold to 3.3 V
			V34	7	Set threshold to 3.4 V
			V35	8	Set threshold to 3.5 V
			V36	9	Set threshold to 3.6 V
			V37	10	Set threshold to 3.7 V
			V38	11	Set threshold to 3.8 V
			V39	12	Set threshold to 3.9 V
			V40	13	Set threshold to 4.0 V
			V41	14	Set threshold to 4.1 V
			V42	15	Set threshold to 4.2 V

5.3.7.8 GPREGRET

Address offset: 0x51C

General purpose retention register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	GPREGRET			General purpose retention register This register is a retained register

5.3.7.9 GPREGRET2

Address offset: 0x520

General purpose retention register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	GPREGRET			General purpose retention register This register is a retained register

5.3.7.10 DCDCEN

Address offset: 0x578

Enable DC/DC converter for REG1 stage.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DCDCEN			Enable DC/DC converter for REG1 stage.
			Disabled	0	Disable
			Enabled	1	Enable

5.3.7.11 DCDCEN0

Address offset: 0x580

Enable DC/DC converter for REG0 stage.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DCDCEN			Enable DC/DC converter for REG0 stage.
			Disabled	0	Disable
			Enabled	1	Enable

5.3.7.12 MAINREGSTATUS

Address offset: 0x640

Main supply status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	MAINREGSTATUS			Main supply status
			Normal	0	Normal voltage mode. Voltage supplied on VDD.
			High	1	High voltage mode. Voltage supplied on VDDH.

5.3.7.13 RAM[n].POWER (n=0..8)

Address offset: 0x900 + (n × 0x10)

RAMn power control register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x0000FFFF				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A-P	RW	S[i]POWER (i=0..15)			Keep RAM section Si on or off in System ON mode. RAM sections are always retained when on, but can also be retained when off depending on the settings in SiRETENTION. All RAM sections will be off in System OFF mode.
			Off	0	Off
			On	1	On
Q-f	RW	S[i]RETENTION (i=0..15)			Keep retention on RAM section Si when RAM section is off
			Off	0	Off
			On	1	On

5.3.7.14 RAM[n].POWERSET (n=0..8)

Address offset: 0x904 + (n × 0x10)

RAMn power control set register

When read, this register will return the value of the POWER register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x0000FFFF				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A-P	W	S[i]POWER (i=0..15)			Keep RAM section Si of RAMn on or off in System ON mode
			On	1	On
Q-f	W	S[i]RETENTION (i=0..15)			Keep retention on RAM section Si when RAM section is switched off
			On	1	On

5.3.7.15 RAM[n].POWERCLR (n=0..8)

Address offset: 0x908 + (n × 0x10)

RAMn power control clear register

When read, this register will return the value of the POWER register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x0000FFFF				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A-P	W	S[i]POWER (i=0..15)			Keep RAM section Si of RAMn on or off in System ON mode
			Off	1	Off
Q-f	W	S[i]RETENTION (i=0..15)			Keep retention on RAM section Si when RAM section is switched off
			Off	1	Off

5.3.8 Electrical specification

5.3.8.1 Regulator operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
$V_{DD,POR}$	VDD supply voltage needed during power-on reset.	1.75			V
V_{DD}	VDD operating voltage.	1.7	3.0	3.6	V
V_{DDH}	VDDH operating voltage.	2.5	3.7	5.5	V
C_{VDD}	Effective decoupling capacitance on the VDD pin.	2.7	4.7	5.5	µF
C_{DEC4}	Effective decoupling capacitance on the DEC4 pin.	0.7	1	1.3	µF

5.3.8.2 Regulator specifications, REG0 stage

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDOUT}	VDD output voltage.	1.8		3.3	V
$V_{DDOUT,ERR}$	VDD output voltage error (deviation from setting in REGOUT0 on page 45).	-10		5	%
$I_{EXT,OFF}$	External current draw[10] allowed in High voltage mode (supply on VDDH) during System OFF.			1	mA
$I_{EXT,LOW}$	External current draw[10] allowed in High voltage mode (supply on VDDH) when radio output power is higher than 4 dBm.			5	mA
$I_{EXT,HIGH}$	External current draw[10] allowed in High voltage mode (supply on VDDH) when radio output power is lower than or equal to 4 dBm.			25	mA
$V_{REG0,DROP}$	Minimum voltage drop in REG0 (difference between voltage supplied on VDDH pin and voltage output on VDD pin).	0.3			V

5.3.8.3 Device startup times

Symbol	Description	Min.	Typ.	Max.	Units
t_{POR}	Time in power-on reset after supply reaches minimum operating voltage, depending on supply rise time				
$t_{POR,10µs}$	VDD rise time 10 µs		1	10	ms
$t_{POR,10ms}$	VDD rise time 10 ms[11].		9		ms
$t_{POR,60ms}$	VDD rise time 60 ms[11].		23	110	ms
$t_{RISE,REG0OUT}$	REG0 output (VDD) rise time after VDDH reaches minimum VDDH supply voltage[11].				
$t_{RISE,REG0OUT,10µs}$	VDDH rise time 10 µs[11].		0.22	1.55	ms
$t_{RISE,REG0OUT,10ms}$	VDDH rise time 10 ms[11].		5		ms
$t_{RISE,REG0OUT,100ms}$	VDDH rise time 100 ms[11].	30	50	80	ms
t_{PINR}	Reset time when using pin reset, depending on pin capacitance	..	..	..	
$t_{PINR,500nF}$	500 nF capacitance at reset pin.			32.5	ms
$t_{PINR,10µF}$	10 µF capacitance at reset pin.			650	ms
t_{R2ON}	Time from power-on reset to System ON.				
$t_{R2ON,NOTCONF}$	If reset pin not configured.	tPOR			ms

[10] External current draw is defined as the sum of all GPIO currents and the current being drawn from VDD.

[11] See Recommended operating conditions on page 544 for more information.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{R2ON,CONF}$	If reset pin configured.	tPOR + tPINR			ms
t_{OFF2ON}	Time from OFF to CPU execute.		16.5		µs
$t_{IDLE2CPU}$	Time from IDLE to CPU execute.		3.0		µs
$t_{EVTSET,CL1}$	Time from HW event to PPI event in Constant Latency System ON mode.		0.0625		µs
$t_{EVTSET,CL0}$	Time from HW event to PPI event in Low Power System ON mode.		0.0625		µs

5.3.8.4 Power fail comparator

Symbol	Description	Min.	Typ.	Max.	Units
$V_{POF,NV}$	Nominal power level warning thresholds (falling supply voltage) in Normal voltage mode (supply on VDD). Levels are configurable between Min. and Max. in 100 mV increments.	1.7		2.8	V
$V_{POF,HV}$	Nominal power level warning thresholds (falling supply voltage) in High voltage mode (supply on VDDH). Levels are configurable in 100 mV increments.	2.7		4.2	V
V_{POFTOL}	Threshold voltage tolerance (applies in both Normal voltage mode and High voltage mode).	-5		5	%
$V_{POFHYST}$	Threshold voltage hysteresis (applies in both Normal voltage mode and High voltage mode).	40	50	60	mV
$V_{BOR,OFF}$	Brownout reset voltage range System OFF mode. Brownout only applies to the voltage on VDD.	1.2		1.62	V
$V_{BOR,ON}$	Brownout reset voltage range System ON mode. Brownout only applies to the voltage on VDD.	1.57	1.6	1.63	V

5.3.8.5 USB operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{BUS}	Supply voltage on VBUS pin	4.35	5	5.5	V
V_{DPDM}	Voltage on D+ and D- lines	VSS - 0.3 V		VUSB33 + 0.3 V	V

5.3.8.6 USB regulator specifications

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,QUIES}$	USB regulator quiescent current drawn from VBUS (USBD enabled)		170		µA
$t_{USBPWRRDY}$	Time from USB enabled to USBPWRRDY event triggered, V_{BUS} supply provided		1		ms
V_{USB33}	On voltage at the USB regulator output (DECUSB pin)	3.0	3.3	3.6	V
$R_{SOURCE,VBUS}$	Maximum source resistance on VBUS, including cable			2	Ω
C_{DECUSB}	Decoupling capacitor on the DECUSB pin	2.35	4.7	5.5	µF

5.3.8.7 VBUS detection specifications

Symbol	Description	Min.	Typ.	Max.	Units
$V_{BUS,DETECT}$	Voltage at which rising VBUS gets reported by USBDETECTED	3.4	4.0	4.3	V
$V_{BUS,REMOVE}$	Voltage at which decreasing VBUS gets reported by USBREMOVED	3.0	3.6	3.9	V

5.4 CLOCK — Clock control

The clock control system can source the system clocks from a range of internal or external high and low frequency oscillators and distribute them to modules based upon a module's individual requirements. Clock distribution is automated and grouped independently by module to limit current consumption in unused branches of the clock tree.

Listed here are the main features for CLOCK:

- 64 MHz on-chip oscillator
- 64 MHz crystal oscillator, using external 32 MHz crystal
- 32.768 kHz +/-500 ppm RC oscillator
- 32.768 kHz crystal oscillator, using external 32.768 kHz crystal
- 32.768 kHz oscillator synthesized from 64 MHz oscillator
- Firmware (FW) override control of crystal oscillator activity for low latency start up
- Automatic internal oscillator and clock control, and distribution for ultra-low power

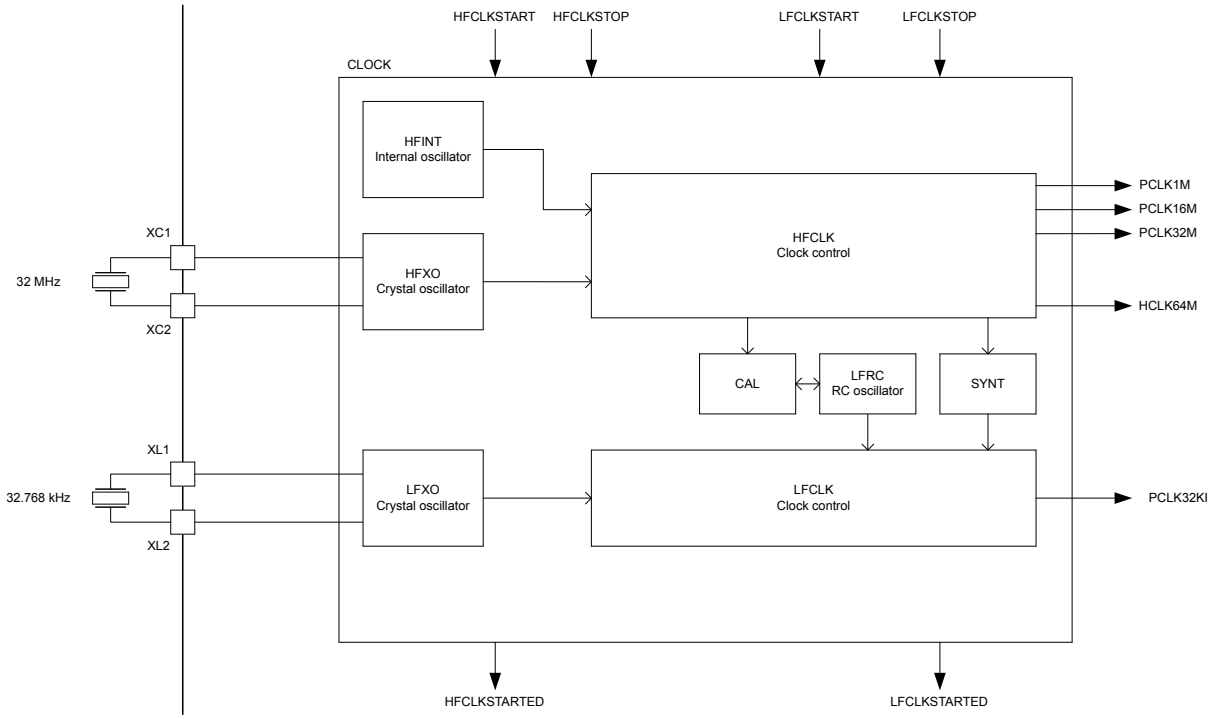

Figure 20: Clock control

5.4.1 HFCLK controller

The HFCLK controller provides several clock signals in the system.

These are as follows:

- HCLK64M: 64 MHz CPU clock
- PCLK1M: 1 MHz peripheral clock
- PCLK16M: 16 MHz peripheral clock
- PCLK32M: 32 MHz peripheral clock

The HFCLK controller uses the following high frequency clock (HFCLK) sources:

- 64 MHz internal oscillator (HFINT)

- 64 MHz crystal oscillator (HFXO)

For illustration, see Clock control on page 80.

The HFCLK controller will automatically provide the clock(s) requested by the system. If the system does not request any clocks from the HFCLK controller, the controller will enter a power saving mode.

The HFINT source will be used when HFCLK is requested and HFXO has not been started.

The HFXO is started by triggering the HFCLKSTART task and stopped by triggering the HFCLKSTOP task. When the HFCLKSTART task is triggered, the HFCLKSTARTED event is generated once the HFXO startup time has elapsed. The HFXO startup time is given as the sum of the following:

- HFXO power-up time, as specified in 64 MHz crystal oscillator (HFXO) on page 90.
- HFXO debounce time, as specified in register HFXODEBOUNCE on page 88.

The HFXO must be running to use the RADIO or the calibration mechanism associated with the 32.768 kHz RC oscillator.

5.4.1.1 64 MHz crystal oscillator (HFXO)

The 64 MHz crystal oscillator (HFXO) is controlled by a 32 MHz external crystal.

The crystal oscillator is designed for use with an AT-cut quartz crystal in parallel resonant mode. To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet.

Circuit diagram of the 64 MHz crystal oscillator on page 81 shows how the 32 MHz crystal is connected to the 64 MHz crystal oscillator.

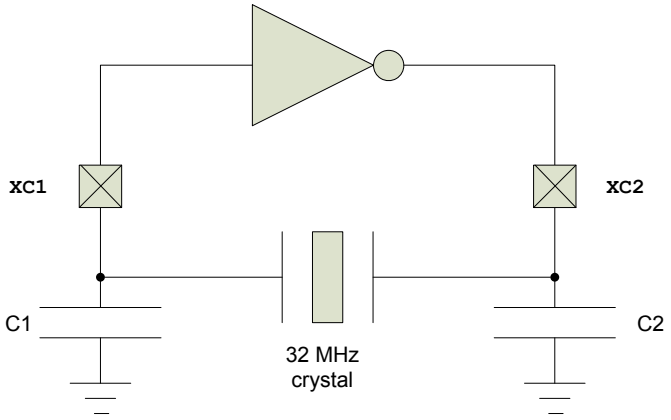

Figure 21: Circuit diagram of the 64 MHz crystal oscillator

The load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is given by:

$$CL = \frac{(C1' \cdot C2')}{(C1' + C2')}$$

$$C1' = C1 + C_{pcb1} + C_{pin}$$
$$C2' = C2 + C_{pcb2} + C_{pin}$$

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. For more information, see Reference circuitry on page 528. C_{pcb1} and C_{pcb2} are stray capacitances on the PCB. C_{pin} is the pin input capacitance on the `XC1` and `XC2` pins. See table 64 MHz crystal oscillator (HFXO) on page 90. The load capacitors C1 and C2 should have the same value.

For reliable operation, the crystal load capacitance, shunt capacitance, equivalent series resistance, and drive level must comply with the specifications in table 64 MHz crystal oscillator (HFXO) on page 90. It is recommended to use a crystal with lower than maximum load capacitance and/or shunt capacitance. A low load capacitance will reduce both start up time and current consumption.

5.4.2 LFCLK controller

The system supports several low frequency clock sources.

As illustrated in Clock control on page 80, the system supports the following low frequency clock sources:

- 32.768 kHz RC oscillator (LFRC)
- 32.768 kHz crystal oscillator (LFXO)
- 32.768 kHz synthesized from HFCLK (LFSYNT)

The LFCLK controller and all of the LFCLK clock sources are always switched off when in System OFF mode.

The LFCLK clock is started by first selecting the preferred clock source in register LFCLKSRC on page 88 and then triggering the LFCLKSTART task. If the LFXO is selected as the clock source, the LFCLK will initially start running from the 32.768 kHz LFRC while the LFXO is starting up and automatically switch to using the LFXO once this oscillator is running. The LFCLKSTARTED event will be generated when the LFXO has been started.

The LFCLK clock is stopped by triggering the LFCLKSTOP task.

Register LFCLKSRC on page 88 controls the clock source, and its allowed swing. The truth table for various situations is as follows:

SRC	EXTERNAL	BYPASS	Comment
0	0	0	Normal operation, LFRC is source
0	0	1	DO NOT USE
0	1	X	DO NOT USE
1	0	0	Normal XTAL operation
1	1	0	Apply external low swing signal to XL1, ground XL2
1	1	1	Apply external full swing signal to XL1, leave XL2 grounded or unconnected
1	0	1	DO NOT USE
2	0	0	Normal operation, LFSYNT is source
2	0	1	DO NOT USE
2	1	X	DO NOT USE

Table 17: LFCLKSRC configuration depending on clock source

It is not allowed to write to register LFCLKSRC on page 88 when the LFCLK is running.

A LFCLKSTOP task will stop the LFCLK oscillator. However, the LFCLKSTOP task can only be triggered after the STATE field in register LFCLKSTAT on page 87 indicates LFCLK running state.

The synthesized 32.768 kHz clock depends on the HFCLK to run. If high accuracy is required for the LFCLK running off the synthesized 32.768 kHz clock, the HFCLK must running from the HFXO source.

5.4.2.1 32.768 kHz RC oscillator (LFRC)

The default source of the low frequency clock (LFCLK) is the 32.768 kHz RC oscillator (LFRC).

The LFRC oscillator has two modes of operation, normal and ultra-low power (ULP) mode, enabling the user to trade power consumption against accuracy of the clock. The LFRC mode is configured in register LFRCMODE on page 90.

The LFRC oscillator has to be stopped before changing the mode of the oscillator.

The LFRC frequency will be affected by variation in temperature. The LFRC oscillator can be calibrated to improve accuracy by using the HFXO as a reference oscillator during calibration. The LFRC oscillator does not require additional external components.

5.4.2.2 Calibrating the 32.768 kHz RC oscillator

After the LFRC oscillator is started and running, it can be calibrated by triggering the CAL task.

The LFRC oscillator will then temporarily request the HFCLK to be used as a reference for the calibration. A DONE event will be generated when calibration has finished. The HFCLK crystal oscillator has to be started (by triggering the HFCLKSTART task) in order for the calibration mechanism to work.

It is not allowed to stop the LFRC or write to LFRCMODE on page 90 during an ongoing calibration.

5.4.2.3 Calibration timer

The calibration timer can be used to time the calibration interval of the 32.768 kHz RC oscillator.

The calibration timer is started by triggering the CTSTART task and stopped by triggering the CTSTOP task. The calibration timer will always start counting down from the value specified in CTIV (Retained) on page 89 and generate a CTTO event when it reaches 0. The calibration timer will automatically stop when it reaches 0.

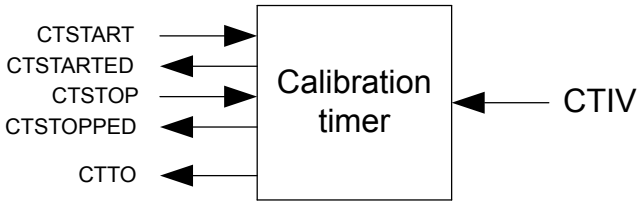

Figure 22: Calibration timer

After a CTSTART task has been triggered, the calibration timer will ignore further tasks until it has returned the CTSTARTED event. Likewise, after a CTSTOP task has been triggered, the calibration timer will ignore further tasks until it has returned a CTSTOPPED event. Triggering CTSTART while the calibration timer is running will immediately return a CTSTARTED event. Triggering CTSTOP when the calibration timer is stopped will immediately return a CTSTOPPED event.

5.4.2.4 32.768 kHz crystal oscillator (LFXO)

For higher LFCLK accuracy (when better than +/- 500 ppm accuracy is required), the low frequency crystal oscillator (LFXO) must be used.

The following external clock sources are supported:

- Low swing clock signal applied to the *XL1* pin. The *XL2* pin shall then be grounded.
- Rail-to-rail clock signal applied to the *XL1* pin. The *XL2* pin shall then be grounded or left unconnected.

To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet. Circuit diagram of the 32.768 kHz crystal oscillator on page 84 shows the LFXO circuitry.

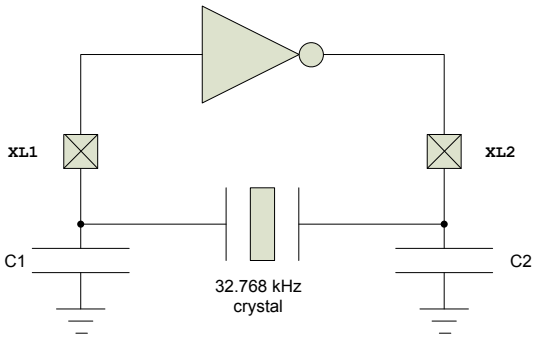

Figure 23: Circuit diagram of the 32.768 kHz crystal oscillator

The load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is given by:

$$CL = \frac{(C1' \cdot C2')}{(C1' + C2')}$$

$$C1' = C1 + C_{pcb1} + C_{pin}$$
$$C2' = C2 + C_{pcb2} + C_{pin}$$

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. C_{pcb1} and C_{pcb2} are stray capacitances on the PCB. C_{pin} is the pin input capacitance on the XC1 and XC2 pins (see Low frequency crystal oscillator (LFXO) on page 91). The load capacitors C1 and C2 should have the same value.

For more information, see Reference circuitry on page 528.

5.4.2.5 32.768 kHz synthesized from HFCLK (LFSYNT)

LFCLK can also be synthesized from the HFCLK clock source. The accuracy of LFCLK will then be the accuracy of the HFCLK.

Using the LFSYNT clock avoids the requirement for a 32.768 kHz crystal, but increases average power consumption as the HFCLK will need to be requested in the system.

5.4.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	CLOCK	CLOCK	Clock control	

Table 18: Instances

Register	Offset	Description
TASKS_HFCLKSTART	0x000	Start HFXO crystal oscillator
TASKS_HFCLKSTOP	0x004	Stop HFXO crystal oscillator
TASKS_LFCLKSTART	0x008	Start LFCLK
TASKS_LFCLKSTOP	0x00C	Stop LFCLK
TASKS_CAL	0x010	Start calibration of LFRC
TASKS_CTSTART	0x014	Start calibration timer
TASKS_CTSTOP	0x018	Stop calibration timer
EVENTS_HFCLKSTARTED	0x100	HFXO crystal oscillator started
EVENTS_LFCLKSTARTED	0x104	LFCLK started

Register	Offset	Description	
EVENTS_DONE	0x10C	Calibration of LFRC completed	
EVENTS_CTTO	0x110	Calibration timer timeout	
EVENTS_CTSTARTED	0x128	Calibration timer has been started and is ready to process new tasks	
EVENTS_CTSTOPPED	0x12C	Calibration timer has been stopped and is ready to process new tasks	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
HFCLKRUN	0x408	Status indicating that HFCLKSTART task has been triggered	
HFCLKSTAT	0x40C	HFCLK status	
LFCLKRUN	0x414	Status indicating that LFCLKSTART task has been triggered	
LFCLKSTAT	0x418	LFCLK status	
LFCLKSRCCOPY	0x41C	Copy of LFCLKSRC register, set when LFCLKSTART task was triggered	
LFCLKSRC	0x518	Clock source for the LFCLK	
HFXODEBOUNCE	0x528	HFXO debounce time. The HFXO is started by triggering the TASKS_HFCLKSTART task.	
CTIV	0x538	Calibration timer interval	Retained
TRACECONFIG	0x55C	Clocking options for the trace port debug interface	
LFRCMODE	0x5B4	LFRC mode configuration	

Table 19: Register overview

5.4.3.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number																															
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																								F	E						D	C		B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value	Description																														
A	RW	HFCLKSTARTED			Write '1' to enable interrupt for HFCLKSTARTED event See EVENTS_HFCLKSTARTED																														
			Set	1	Enable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
B	RW	LFCLKSTARTED			Write '1' to enable interrupt for LFCLKSTARTED event See EVENTS_LFCLKSTARTED																														
			Set	1	Enable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
C	RW	DONE			Write '1' to enable interrupt for DONE event See EVENTS_DONE																														
			Set	1	Enable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
D	RW	CTTO			Write '1' to enable interrupt for CTTO event See EVENTS_CTTO																														
			Set	1	Enable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
E	RW	CTSTARTED			Write '1' to enable interrupt for CTSTARTED event See EVENTS_CTSTARTED																														
			Set	1	Enable																														

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	CTSTOPPED			Write '1' to enable interrupt for CTSTOPPED event See EVENTS_CTSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.4.3.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	HFCLKSTARTED			Write '1' to disable interrupt for HFCLKSTARTED event See EVENTS_HFCLKSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	LFCLKSTARTED			Write '1' to disable interrupt for LFCLKSTARTED event See EVENTS_LFCLKSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to disable interrupt for DONE event See EVENTS_DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CTTO			Write '1' to disable interrupt for CTTO event See EVENTS_CTTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CTSTARTED			Write '1' to disable interrupt for CTSTARTED event See EVENTS_CTSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	CTSTOPPED			Write '1' to disable interrupt for CTSTOPPED event See EVENTS_CTSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.4.3.3 HFCLKRUN

Address offset: 0x408

Status indicating that HFCLKSTART task has been triggered

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	STATUS			HFCLKSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

5.4.3.4 HFCLKSTAT

Address offset: 0x40C

HFCLK status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	SRC			Source of HFCLK
			RC	0	64 MHz internal oscillator (HFINT)
			Xtal	1	64 MHz crystal oscillator (HFXO)
B	R	STATE			HFCLK state
			NotRunning	0	HFCLK not running
			Running	1	HFCLK running

5.4.3.5 LFCLKRUN

Address offset: 0x414

Status indicating that LFCLKSTART task has been triggered

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	STATUS			LFCLKSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

5.4.3.6 LFCLKSTAT

Address offset: 0x418

LFCLK status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	SRC			Source of LFCLK
			RC	0	32.768 kHz RC oscillator (LFRC)
			Xtal	1	32.768 kHz crystal oscillator (LFXO)
			Synth	2	32.768 kHz synthesized from HFCLK (LFSYNT)
B	R	STATE			LFCLK state
			NotRunning	0	LFCLK not running
			Running	1	LFCLK running

5.4.3.7 LFCLKSRCCOPY

Address offset: 0x41C

Copy of LFCLKSRC register, set when LFCLKSTART task was triggered

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	SRC			Clock source
			RC	0	32.768 kHz RC oscillator (LFRC)
			Xtal	1	32.768 kHz crystal oscillator (LFXO)
			Synth	2	32.768 kHz synthesized from HFCLK (LFSYNT)

5.4.3.8 LFCLKSRC

Address offset: 0x518

Clock source for the LFCLK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SRC			Clock source
			RC	0	32.768 kHz RC oscillator (LFRC)
			Xtal	1	32.768 kHz crystal oscillator (LFXO)
			Synth	2	32.768 kHz synthesized from HFCLK (LFSYNT)
B	RW	BYPASS			Enable or disable bypass of LFCLK crystal oscillator with external clock source
			Disabled	0	Disable (use with Xtal or low-swing external source)
			Enabled	1	Enable (use with rail-to-rail external source)
C	RW	EXTERNAL			Enable or disable external source for LFCLK
			Disabled	0	Disable external source (use with Xtal)
			Enabled	1	Enable use of external source instead of Xtal (SRC needs to be set to Xtal)

5.4.3.9 HFXODEBOUNCE

Address offset: 0x528

HFXO debounce time. The HFXO is started by triggering the TASKS_HFCLKSTART task.

The EVENTS_HFCLKSTARTED event is generated after the HFXO power up time + the HFXO debounce time has elapsed. It is not allowed to change the value of this register while the HFXO is starting.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000010				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	HFXODEBOUNCE		0x01..0xFF	HFXO debounce time. Debounce time = HFXODEBOUNCE * 16 us.
			Db256us	0x10	256 us debounce time. Recommended for TSX-3225, FA-20H and FA-128 crystals.
			Db1024us	0x40	1024 us debounce time. Recommended for NX1612AA and NX1210AB crystals.

5.4.3.10 CTIV (Retained)

Address offset: 0x538

This register is a retained register

Calibration timer interval

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CTIV			Calibration timer interval in multiple of 0.25 seconds. Range: 0.25 seconds to 31.75 seconds.

5.4.3.11 TRACECONFIG

Address offset: 0x55C

Clocking options for the trace port debug interface

This register is a retained register. Reset behavior is the same as debug components.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TRACEPORTSPEED			Speed of trace port clock. Note that the TRACECLK pin will output this clock divided by two.
			32MHz	0	32 MHz trace port clock (TRACECLK = 16 MHz)
			16MHz	1	16 MHz trace port clock (TRACECLK = 8 MHz)
			8MHz	2	8 MHz trace port clock (TRACECLK = 4 MHz)
			4MHz	3	4 MHz trace port clock (TRACECLK = 2 MHz)
B	RW	TRACEMUX			Pin multiplexing of trace signals. See pin assignment chapter for more details.
			GPIO	0	No trace signals routed to pins. All pins can be used as regular GPIOs.
			Serial	1	SWO trace signal routed to pin. Remaining pins can be used as regular GPIOs.
			Parallel	2	All trace signals (TRACECLK and TRACEDATA[n]) routed to pins.

5.4.3.12 LFRCMODE

Address offset: 0x5B4

LFRC mode configuration

Bit number						31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID						B A
Reset 0x00000000						**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description	
A	RW	MODE			Set LFRC mode	
			Normal	0	Normal mode	
			ULP	1	Ultra-low power mode (ULP)	
B	RW	STATUS			Active LFRC mode. This field is read only.	
			Normal	0	Normal mode	
			ULP	1	Ultra-low power mode (ULP)	

5.4.4 Electrical specification

5.4.4.1 64 MHz internal oscillator (HFINT)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_HFINT}	Nominal output frequency		64		MHz
f_{TOL_HFINT}	Frequency tolerance		±1.5	±8	%

5.4.4.2 64 MHz crystal oscillator (HFXO)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_HFXO}	Nominal output frequency		64		MHz
f_{XTAL_HFXO}	External crystal frequency		32		MHz
f_{TOL_HFXO}	Frequency tolerance requirement for 2.4 GHz proprietary radio applications			±60	ppm
$f_{TOL\_HFXO\_BLE}$	Frequency tolerance requirement, Bluetooth low energy applications, packet length <= 200 bytes			±40	ppm
$f_{TOL\_HFXO\_BLE\_LP}$	Frequency tolerance requirement, Bluetooth low energy applications, packet length > 200 bytes			±30	ppm
C_{L_HFXO}	Load capacitance			12	pF
C_{0_HFXO}	Shunt capacitance			7	pF
$R_{S\_HFXO\_7PF}$	Equivalent series resistance 3 pF < C0 <= 7 pF			60	Ω
$R_{S\_HFXO\_3PF}$	Equivalent series resistance C0 <= 3 pF			100	Ω
P_{D_HFXO}	Drive level			100	µW
C_{PIN_HFXO}	Input capacitance XC1 and XC2		3		pF
I_{STBY_X32M}	Core standby current for various crystals				
$I_{STBY\_X32M\_X0}$	Epson TSX-3225		80		µA
$I_{STBY\_X32M\_X1}$	Epson FA-20H		72		µA
$I_{STBY\_X32M\_X2}$	Epson FA-128		70		µA
$I_{STBY\_X32M\_X3}$	NDK NX1612AA		136		µA
$I_{STBY\_X32M\_X4}$	NDK NX1210AB		143		µA
I_{START_X32M}	Average startup current for various crystals, first 1 ms				
$I_{START\_X32M\_X0}$	Epson TSX-3225		328		µA
$I_{START\_X32M\_X1}$	Epson FA-20H		363		µA
$I_{START\_X32M\_X2}$	Epson FA-128		396		µA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{START\_X32M\_X3}$	NDK NX1612AA		783		µA
$I_{START\_X32M\_X4}$	NDK NX1210AB		833		µA
t_{POWER_X32M}	Power-up time for various crystals				
$t_{POWER\_X32M\_X0}$	Epson TSX-3225		50		µs
$t_{POWER\_X32M\_X1}$	Epson FA-20H		60		µs
$t_{POWER\_X32M\_X2}$	Epson FA-128		75		µs
$t_{POWER\_X32M\_X3}$	NDK NX1612AA		195		µs
$t_{POWER\_X32M\_X4}$	NDK NX1210AB		210		µs

5.4.4.3 Low frequency crystal oscillator (LFXO)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFXO}	Crystal frequency		32.768		kHz
$f_{TOL\_LFXO\_BLE}$	Frequency tolerance requirement for BLE stack			±500	ppm
$f_{TOL\_LFXO\_ANT}$	Frequency tolerance requirement for ANT stack			±50	ppm
C_{L_LFXO}	Load capacitance			12.5	pF
C_{0_LFXO}	Shunt capacitance			2	pF
R_{S_LFXO}	Equivalent series resistance			100	kΩ
P_{D_LFXO}	Drive level			1	µW
C_{pin}	Input capacitance on XL1 and XL2 pads		4		pF
I_{LFXO}	Run current for 32.768 kHz crystal oscillator		0.23		µA
t_{START_LFXO}	Startup time for 32.768 kHz crystal oscillator		0.25		s

5.4.4.4 Low frequency RC oscillator (LFRC), Normal mode

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFRC}	Nominal frequency		32.768		kHz
f_{TOL_LFRC}	Frequency tolerance, uncalibrated			±5	%
$f_{TOL\_CAL\_LFRC}$	Frequency tolerance after calibration[12]			±500	ppm
I_{LFRC}	Run current		0.7		µA
t_{START_LFRC}	Startup time		1000		µs

5.4.4.5 Low frequency RC oscillator (LFRC), Ultra-low power mode (ULP)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFULP}	Nominal frequency		32.768		kHz
$f_{TOL\_UNCAL\_LFULP}$	Frequency tolerance, uncalibrated			±7	%
$f_{TOL\_CAL\_LFULP}$	Frequency tolerance after calibration[13]			±2000	ppm
I_{LFULP}	Run current		0.3		µA
t_{START_LFULP}	Startup time		1500		µs

5.4.4.6 Synthesized low frequency clock (LFSYNT)

[12] Constant temperature within ±0.5 °C, calibration performed at least every 8 seconds, averaging interval > 7.5 ms, defined as 3 sigma

[13] Constant temperature within ±0.5 °C, calibration performed at least every 8 seconds, averaging interval > 125 ms, defined as 3 sigma

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFSYNT}	Nominal frequency		32.768		kHz

6 Peripherals

6.1 Peripheral interface

Peripherals are controlled by the CPU by writing to configuration registers and task registers. Peripheral events are indicated to the CPU by event registers and interrupts if they are configured for a given event.

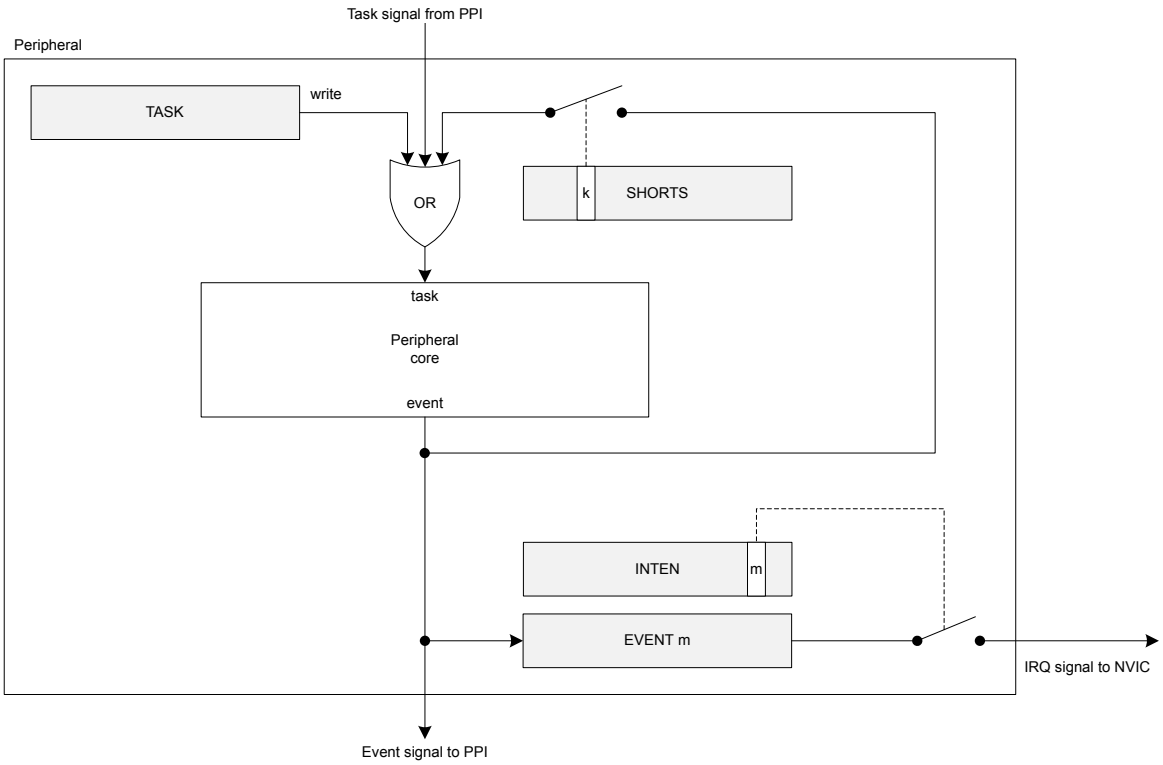

Figure 24: Tasks, events, shortcuts, and interrupts

6.1.1 Peripheral ID

Every peripheral is assigned a fixed block of 0x1000 bytes of address space, which is equal to 1024 x 32 bit registers.

See Instantiation on page 22 for more information about which peripherals are available and where they are located in the address map.

There is a direct relationship between peripheral ID and base address. For example, a peripheral with base address 0x40000000 is assigned ID=0, a peripheral with base address 0x40001000 is assigned ID=1, and a peripheral with base address 0x4001F000 is assigned ID=31.

Peripherals may share the same ID, which may impose one or more of the following limitations:

- Some peripherals share some registers or other common resources.
- Operation is mutually exclusive. Only one of the peripherals can be used at a time.
- Switching from one peripheral to another must follow a specific pattern (disable the first, then enable the second peripheral).

6.1.2 Peripherals with shared ID

In general (with the exception of ID 0), peripherals sharing an ID and base address may not be used simultaneously. The user can only enable one peripheral at the time on this specific ID.

When switching between two peripherals sharing an ID, the user should do the following to prevent unwanted behavior:

- Disable the previously used peripheral.
- Remove any programmable peripheral interconnect (PPI) connections set up for the peripheral that is being disabled.
- Clear all bits in the INTEN register, i.e. INTENCLR = 0xFFFFFFFF.
- Explicitly configure the peripheral that you are about to enable and do not rely on configuration values that may be inherited from the peripheral that was disabled.
- Enable the now configured peripheral.

See which peripherals are sharing ID in Instantiation on page 22.

6.1.3 Peripheral registers

Most peripherals feature an ENABLE register. Unless otherwise specified in the relevant chapter, the peripheral registers (in particular the PSEL registers) must be configured before enabling the peripheral.

Note that the peripheral must be enabled before tasks and events can be used.

6.1.4 Bit set and clear

Registers with multiple single-bit bit fields may implement the set-and-clear pattern. This pattern enables firmware to set and clear individual bits in a register without having to perform a read-modify-write operation on the main register.

This pattern is implemented using three consecutive addresses in the register map, where the main register is followed by dedicated SET and CLR registers (in that exact order).

The SET register is used to set individual bits in the main register while the CLR register is used to clear individual bits in the main register. Writing `1` to a bit in SET or CLR register will set or clear the same bit in the main register respectively. Writing `0` to a bit in SET or CLR register has no effect. Reading the SET or CLR register returns the value of the main register.

Note: The main register may not be visible and hence not directly accessible in all cases.

6.1.5 Tasks

Tasks are used to trigger actions in a peripheral, for example to start a particular behavior. A peripheral can implement multiple tasks with each task having a separate register in that peripheral's task register group.

A task is triggered when firmware writes `1` to the task register, or when the peripheral itself or another peripheral toggles the corresponding task signal. See Tasks, events, shortcuts, and interrupts on page 93.

6.1.6 Events

Events are used to notify peripherals and the CPU about events that have happened, for example a state change in a peripheral. A peripheral may generate multiple events with each event having a separate register in that peripheral's event register group.

An event is generated when the peripheral itself toggles the corresponding event signal, and the event register is updated to reflect that the event has been generated. See Tasks, events, shortcuts, and interrupts on page 93. An event register is only cleared when firmware writes `0` to it.

Events can be generated by the peripheral even when the event register is set to `1`.

6.1.7 Shortcuts

A shortcut is a direct connection between an event and a task within the same peripheral. If a shortcut is enabled, the associated task is automatically triggered when its associated event is generated.

Using a shortcut is the equivalent to making the same connection outside the peripheral and through the PPI. However, the propagation delay through the shortcut is usually shorter than the propagation delay through the PPI.

Shortcuts are predefined, which means their connections cannot be configured by firmware. Each shortcut can be individually enabled or disabled through the shortcut register, one bit per shortcut, giving a maximum of 32 shortcuts for each peripheral.

6.1.8 Interrupts

All peripherals support interrupts. Interrupts are generated by events.

A peripheral only occupies one interrupt, and the interrupt number follows the peripheral ID. For example, the peripheral with ID=4 is connected to interrupt number 4 in the nested vectored interrupt controller (NVIC).

Using the INTEN, INTENSET and INTENCLR registers, every event generated by a peripheral can be configured to generate that peripheral's interrupt. Multiple events can be enabled to generate interrupts simultaneously. To resolve the correct interrupt source, the event registers in the event group of peripheral registers will indicate the source.

Some peripherals implement only INTENSET and INTENCLR registers, and the INTEN register is not available on those peripherals. See the individual peripheral chapters for details. In all cases, reading back the INTENSET or INTENCLR register returns the same information as in INTEN.

Each event implemented in the peripheral is associated with a specific bit position in the INTEN, INTENSET and INTENCLR registers.

The relationship between tasks, events, shortcuts, and interrupts is shown in Tasks, events, shortcuts, and interrupts on page 93.

Interrupt clearing

Clearing an interrupt by writing `0` to an event register, or disabling an interrupt using the INTENCLR register, can take up to four CPU clock cycles to take effect. This means that an interrupt may reoccur immediatelly, even if a new event has not come, if the program exits an interrupt handler after the interrupt is cleared or disabled but before four clock cycles have passed.

> **Note:** To avoid an interrupt reoccurring before a new event has come, the program should perform a read from one of the peripheral registers. For example, the event register that has been cleared, or the INTENCLR register that has been used to disable the interrupt. This will cause a one to three-cycle delay and ensure the interrupt is cleared before exiting the interrupt handler.

Care should be taken to ensure the compiler does not remove the read operation as an optimization. If the program can guarantee a four-cycle delay after event being cleared or interrupt disabled in any other way, then a read of a register is not required.

6.2 AAR — Accelerated address resolver

Accelerated address resolver is a cryptographic support function for implementing the "Resolvable Private Address Resolution Procedure" described in the *Bluetooth Core specification* v4.0. "Resolvable private address generation" should be achieved using ECB and is not supported by AAR.

The procedure allows two devices that share a secret key to generate and resolve a hash based on their device address. The AAR block enables real-time address resolution on incoming packets when configured as described in this chapter. This allows real-time packet filtering (whitelisting) using a list of known shared keys (Identity Resolving Keys (IRK) in *Bluetooth*).

6.2.1 EasyDMA

The AAR implements EasyDMA for reading and writing to the RAM. The EasyDMA will have finished accessing the RAM when the END, RESOLVED, and NOTRESOLVED events are generated.

If the IRKPTR on page 100, ADDRPTR on page 100 and the SCRATCHPTR on page 100 is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

6.2.2 Resolving a resolvable address

As per *Bluetooth* specification, a private resolvable address is composed of six bytes.

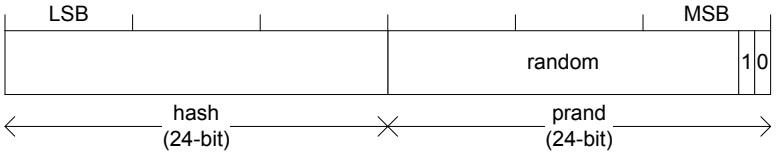

Figure 25: Resolvable address

To resolve an address the ADDRPTR on page 100 register must point to the start of packet. The resolver is started by triggering the START task. A RESOLVED event is generated when the AAR manages to resolve the address using one of the Identity Resolving Keys (IRK) found in the IRK data structure. The AAR will use the IRK specified in the register IRK0 to IRK15 starting from IRK0. How many to be used is specified by the NIRK register. The AAR module will generate a NOTRESOLVED event if it is not able to resolve the address using the specified list of IRKs.

The AAR will go through the list of available IRKs in the IRK data structure and for each IRK try to resolve the address according to the Resolvable Private Address Resolution Procedure described in the *Bluetooth* Specification[14]. The time it takes to resolve an address may vary depending on where in the list the resolvable address is located. The resolution time will also be affected by RAM accesses performed by other peripherals and the CPU. See the Electrical specifications for more information about resolution time.

The AAR will only do a comparison of the received address to those programmed in the module. And not check what type of address it actually is.

The AAR will stop as soon as it has managed to resolve the address, or after trying to resolve the address using NIRK number of IRKs from the IRK data structure. The AAR will generate an END event after it has stopped.

[14] *Bluetooth* Specification Version 4.0 [Vol 3] chapter 10.8.2.3.

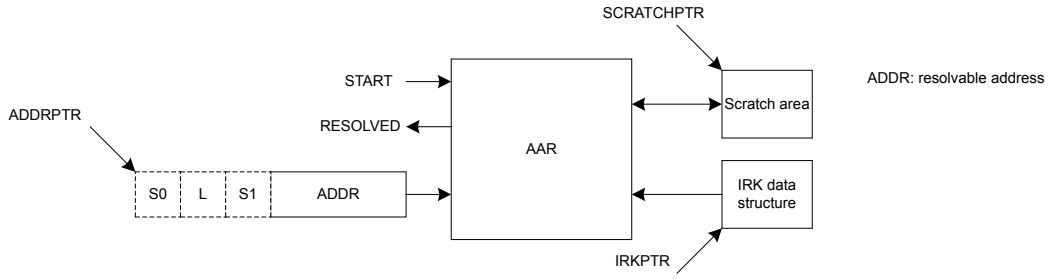

Figure 26: Address resolution with packet preloaded into RAM

6.2.3 Use case example for chaining RADIO packet reception with address resolution using AAR

The AAR may be started as soon as the 6 bytes required by the AAR have been received by the RADIO and stored in RAM. The ADDRPTR pointer must point to the start of packet.

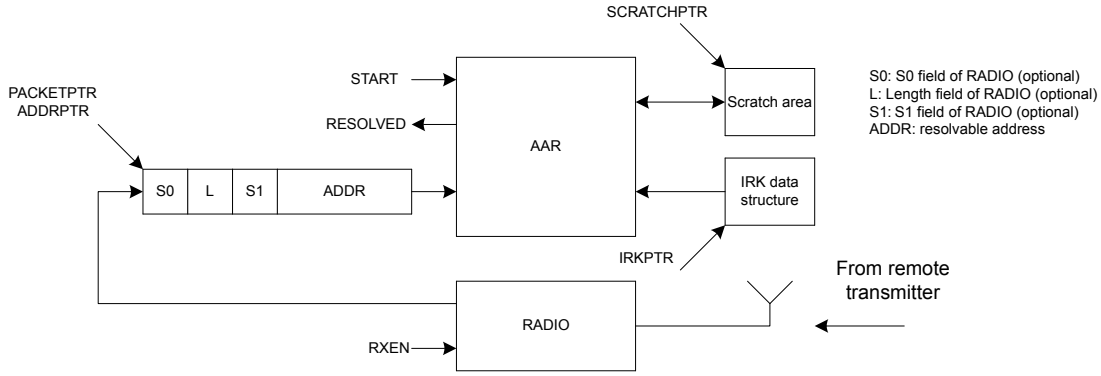

Figure 27: Address resolution with packet loaded into RAM by the RADIO

6.2.4 IRK data structure

The IRK data structure is located in RAM at the memory location specified by the IRKPTR register.

Property	Address offset	Description
IRK0	0	IRK number 0 (16 - byte)
IRK1	16	IRK number 1 (16 - byte)
..	..	..
IRK15	240	IRK number 15 (16 - byte)

Table 20: IRK data structure overview

6.2.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	AAR	AAR	Accelerated address resolver	

Table 21: Instances

Register	Offset	Description
TASKS_START	0x000	Start resolving addresses based on IRKs specified in the IRK data structure
TASKS_STOP	0x008	Stop resolving addresses
EVENTS_END	0x100	Address resolution procedure complete

Register	Offset	Description
EVENTS_RESOLVED	0x104	Address resolved
EVENTS_NOTRESOLVED	0x108	Address not resolved
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
STATUS	0x400	Resolution status
ENABLE	0x500	Enable AAR
NIRK	0x504	Number of IRKs
IRKPTR	0x508	Pointer to IRK data structure
ADDRPTR	0x510	Pointer to the resolvable address
SCRATCHPTR	0x514	Pointer to data area used for temporary storage

Table 22: Register overview

6.2.5.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	RESOLVED			Write '1' to enable interrupt for RESOLVED event See EVENTS_RESOLVED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	NOTRESOLVED			Write '1' to enable interrupt for NOTRESOLVED event See EVENTS_NOTRESOLVED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.2.5.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END			Write '1' to disable interrupt for END event See EVENTS_END
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
B	RW	RESOLVED			Write '1' to disable interrupt for RESOLVED event See EVENTS_RESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	NOTRESOLVED			Write '1' to disable interrupt for NOTRESOLVED event See EVENTS_NOTRESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.2.5.3 STATUS

Address offset: 0x400

Resolution status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	STATUS		[0..15]	The IRK that was used last time an address was resolved

6.2.5.4 ENABLE

Address offset: 0x500

Enable AAR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable AAR
			Disabled	0	Disable
			Enabled	3	Enable

6.2.5.5 NIRK

Address offset: 0x504

Number of IRKs

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	RW	Field	Value ID	Value	Description
A	RW	NIRK		[1..16]	Number of Identity root keys available in the IRK data structure

6.2.5.6 IRKPTR

Address offset: 0x508

Pointer to IRK data structure

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	IRKPTR			Pointer to the IRK data structure

6.2.5.7 ADDRPTR

Address offset: 0x510

Pointer to the resolvable address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ADDRPTR			Pointer to the resolvable address (6-bytes)

6.2.5.8 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SCRATCHPTR			Pointer to a scratch data area used for temporary storage during resolution.A space of minimum 3 bytes must be reserved.

6.2.6 Electrical specification

6.2.6.1 AAR Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{AAR}	Address resolution time per IRK. Total time for several IRKs is given as (1 µs + n * t_AAR), where n is the number of IRKs. (Given priority to the actual destination RAM block).			6	µs
$t_{AAR,8}$	Time for address resolution of 8 IRKs. (Given priority to the actual destination RAM block).			49	µs

6.3 ACL — Access control lists

The Access control lists (ACL) peripheral is designed to assign and enforce access permissions to different regions of the on-chip flash memory map.

Flash memory regions can be assigned individual ACL permission schemes. The following registers are involved:

- PERM register, where the permissions are configured.
- ADDR register, where the word-aligned start address for the flash page is defined.
- SIZE register, where the size of the region the permissions are applied to is determined.

Important: The size of the region in bytes is restricted to a multiple of the flash page size. See the Memory on page 19 chapter for more information.

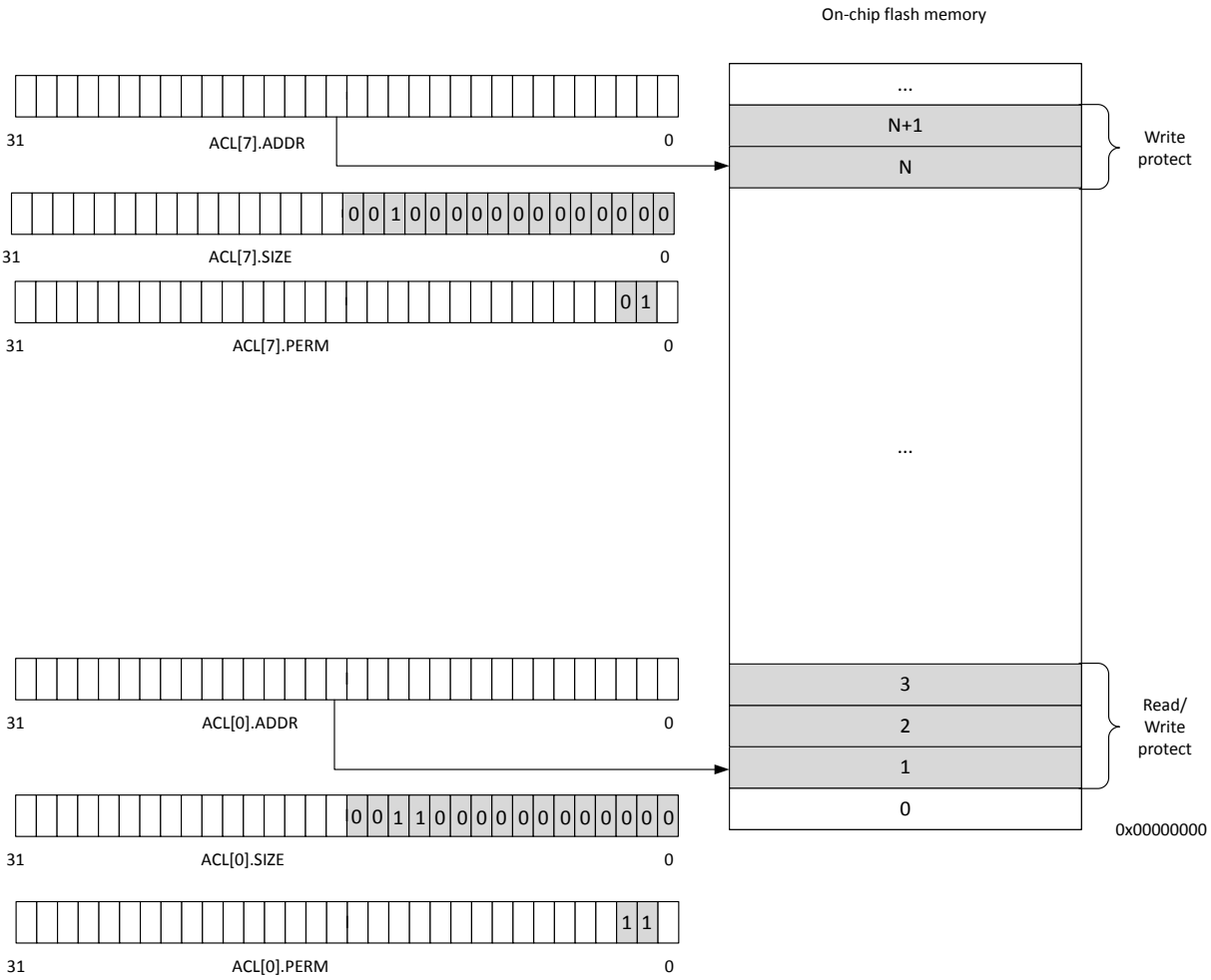

Figure 28: Protected regions of on-chip flash memory

There are four defined ACL permission schemes, with different combinations of read/write permissions:

Read	Write	Protection description
0	0	No protection. Entire region can be executed, read, written or erased.
0	1	Region can be executed and read, but not written or erased.
1	0	Region can be written and erased, but not executed or read.
1	1	Region is locked for all access until next reset.

Table 23: Permission schemes

Important: If a permission violation to a protected region is detected by the ACL peripheral, the request is blocked and a Bus Fault exception is triggered.

Access control to a configured region is enforced by the hardware two CPU clock cycles after the ADDR, SIZE, and PERM registers for an ACL instance have been successfully written. The protection is only enforced if a valid start address of the flash page boundary is written into the ADDR register, and the values of the SIZE and PERM registers are not zero.

The ADDR, SIZE, and PERM registers can only be written once. All ACL configuration registers are cleared on reset (by resetting the device from any reset source), which is also the only way of clearing the configuration registers. To ensure that the desired permission schemes are always enforced by the ACL peripheral, the device boot sequence must perform the necessary configuration.

Debugger read access to a read-protected region will be Read-As-Zero (RAZ), while debugger write access to a write-protected region will be Write-Ignored (WI).

6.3.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001E000	ACL	ACL	Access control lists	

Table 24: Instances

Register	Offset	Description	
ACL[0].ADDR	0x800	Configure the word-aligned start address of region 0 to protect	
ACL[0].SIZE	0x804	Size of region to protect counting from address ACL[0].ADDR. Write '0' as no effect.	
ACL[0].PERM	0x808	Access permissions for region 0 as defined by start address ACL[0].ADDR and size ACL[0].SIZE	
ACL[0].UNUSED0	0x80C		Reserved
ACL[1].ADDR	0x810	Configure the word-aligned start address of region 1 to protect	
ACL[1].SIZE	0x814	Size of region to protect counting from address ACL[1].ADDR. Write '0' as no effect.	
ACL[1].PERM	0x818	Access permissions for region 1 as defined by start address ACL[1].ADDR and size ACL[1].SIZE	
ACL[1].UNUSED0	0x81C		Reserved
ACL[2].ADDR	0x820	Configure the word-aligned start address of region 2 to protect	
ACL[2].SIZE	0x824	Size of region to protect counting from address ACL[2].ADDR. Write '0' as no effect.	
ACL[2].PERM	0x828	Access permissions for region 2 as defined by start address ACL[2].ADDR and size ACL[2].SIZE	
ACL[2].UNUSED0	0x82C		Reserved
ACL[3].ADDR	0x830	Configure the word-aligned start address of region 3 to protect	
ACL[3].SIZE	0x834	Size of region to protect counting from address ACL[3].ADDR. Write '0' as no effect.	
ACL[3].PERM	0x838	Access permissions for region 3 as defined by start address ACL[3].ADDR and size ACL[3].SIZE	
ACL[3].UNUSED0	0x83C		Reserved
ACL[4].ADDR	0x840	Configure the word-aligned start address of region 4 to protect	
ACL[4].SIZE	0x844	Size of region to protect counting from address ACL[4].ADDR. Write '0' as no effect.	
ACL[4].PERM	0x848	Access permissions for region 4 as defined by start address ACL[4].ADDR and size ACL[4].SIZE	
ACL[4].UNUSED0	0x84C		Reserved
ACL[5].ADDR	0x850	Configure the word-aligned start address of region 5 to protect	
ACL[5].SIZE	0x854	Size of region to protect counting from address ACL[5].ADDR. Write '0' as no effect.	
ACL[5].PERM	0x858	Access permissions for region 5 as defined by start address ACL[5].ADDR and size ACL[5].SIZE	
ACL[5].UNUSED0	0x85C		Reserved
ACL[6].ADDR	0x860	Configure the word-aligned start address of region 6 to protect	
ACL[6].SIZE	0x864	Size of region to protect counting from address ACL[6].ADDR. Write '0' as no effect.	
ACL[6].PERM	0x868	Access permissions for region 6 as defined by start address ACL[6].ADDR and size ACL[6].SIZE	
ACL[6].UNUSED0	0x86C		Reserved
ACL[7].ADDR	0x870	Configure the word-aligned start address of region 7 to protect	
ACL[7].SIZE	0x874	Size of region to protect counting from address ACL[7].ADDR. Write '0' as no effect.	
ACL[7].PERM	0x878	Access permissions for region 7 as defined by start address ACL[7].ADDR and size ACL[7].SIZE	

Register	Offset	Description	
ACL[7].UNUSED0	0x87C		Reserved

Table 25: Register overview

6.3.1.1 ACL[n].ADDR (n=0..7)

Address offset: 0x800 + (n × 0x10)

Configure the word-aligned start address of region n to protect

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	ADDR			Valid word-aligned start address of region n to protect. Address must point to a flash page boundary.

6.3.1.2 ACL[n].SIZE (n=0..7)

Address offset: 0x804 + (n × 0x10)

Size of region to protect counting from address ACL[n].ADDR. Write '0' as no effect.

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	SIZE			Size of flash region n in bytes. Must be a multiple of the flash page size, and the maximum region size is limited to 512kB.

6.3.1.3 ACL[n].PERM (n=0..7)

Address offset: 0x808 + (n × 0x10)

Access permissions for region n as defined by start address ACL[n].ADDR and size ACL[n].SIZE

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																														C	B	
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
B	RW	WRITE			Configure write and erase permissions for region n. Write '0' has no effect.
			Enable	0	Allow write and erase instructions to region n
			Disable	1	Block write and erase instructions to region n
C	RW	READ			Configure read permissions for region n. Write '0' has no effect.
			Enable	0	Allow read instructions to region n
			Disable	1	Block read instructions to region n

6.4 CCM — AES CCM mode encryption

Cipher block chaining - message authentication code (CCM) mode is an authenticated encryption algorithm designed to provide both authentication and confidentiality during data transfer. CCM combines counter mode encryption and CBC-MAC authentication. The CCM terminology "Message authentication code (MAC)" is called the "Message integrity check (MIC)" in *Bluetooth* terminology and also in this document.

The CCM block generates an encrypted keystream that is applied to input data using the XOR operation and generates the 4 byte MIC field in one operation. The CCM and radio can be configured to work synchronously. The CCM will encrypt in time for transmission and decrypt after receiving bytes into memory from the radio. All operations can complete within the packet RX or TX time. CCM on this device is implemented according to *Bluetooth* requirements and the algorithm as defined in IETF RFC3610, and depends on the AES-128 block cipher. A description of the CCM algorithm can also be found in NIST Special Publication 800-38C. The *Bluetooth* specification describes the configuration of counter mode blocks and encryption blocks to implement compliant encryption for BLE.

The CCM block uses EasyDMA to load key, counter mode blocks (including the nonce required), and to read/write plain text and cipher text.

The AES CCM supports three operations: key-stream generation, packet encryption, and packet decryption. All these operations are done in compliance with the *Bluetooth* specification.[15]

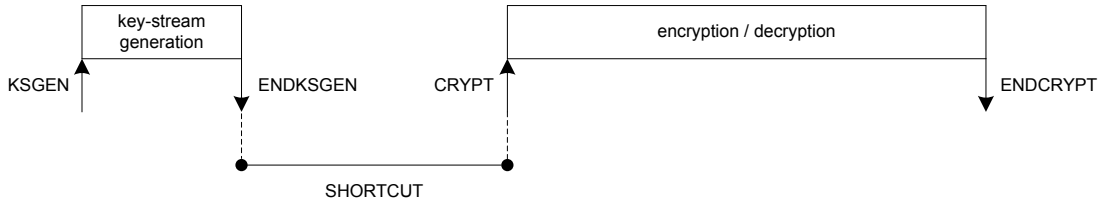

Figure 29: Key-stream generation followed by encryption or decryption. The shortcut is optional.

6.4.1 Key-steam generation

A new key-stream needs to be generated before a new packet encryption or packet decryption operation can be started.

A key-stream is generated by triggering the KSGEN task and an ENDKSGEN event will be generated when the key-stream has been generated.

Key-stream generation, packet encryption, and packet decryption operations utilize the configuration specified in the data structure pointed to by CNFPTR on page 112. It is necessary to configure this pointer and its underlying data structure, and the MODE on page 111 register before the KSGEN task is triggered.

The key-stream will be stored in the AES CCM's temporary memory area, specified by the SCRATCHPTR on page 113, where it will be used in subsequent encryption and decryption operations.

For default length packets (MODE.LENGTH = Default) the size of the generated key-stream is 27 bytes. When using extended length packets (MODE.LENGTH = Extended) the MAXPACKETSIZE on page 113 register specifies the length of the key-stream to be generated. The length of the generated key-stream must be greater or equal to the length of the subsequent packet payload to be encrypted or decrypted. The maximum length of the key-stream in extended mode is 251 bytes, which means that the maximum packet payload size is 251.

[15] *Bluetooth* AES CCM 128 bit block encryption, see *Bluetooth* Core specification Version 4.0.

If a shortcut is used between ENDKSGEN event and CRYPT task, the INPTR on page 112 pointer and the OUTPTR on page 112 pointers must also be configured before the KSGEN task is triggered.

6.4.2 Encryption

During packet encryption, the AES CCM will read the unencrypted packet located in RAM at the address specified in the INPTR pointer, encrypt the packet and append a four byte long Message Integrity Check (MIC) field to the packet.

Encryption is started by triggering the CRYPT task with the MODE on page 111 register set to ENCRYPTION. An ENDCRYPT event will be generated when packet encryption is completed

The AES CCM will also modify the length field of the packet to adjust for the appended MIC field, that is, add four bytes to the length, and store the resulting packet back into RAM at the address specified in the OUTPTR on page 112 pointer, see Encryption on page 105.

Empty packets (length field is set to 0) will not be encrypted but instead moved unmodified through the AES CCM.

The CCM supports different widths of the LENGTH field in the data structure for encrypted packets. This is configured in the MODE on page 111 register.

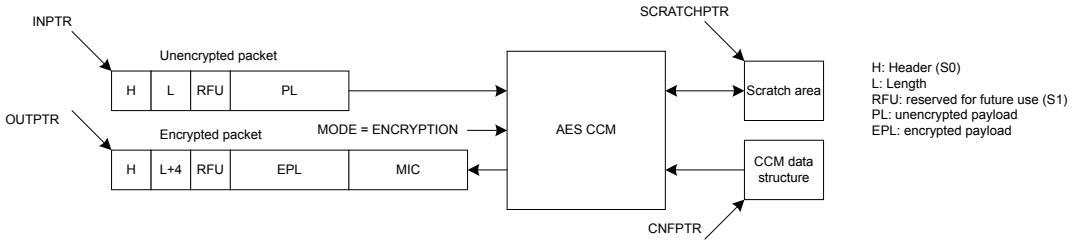

Figure 30: Encryption

6.4.3 Decryption

During packet decryption, the AES CCM will read the encrypted packet located in RAM at the address specified in the INPTR pointer, decrypt the packet, authenticate the packet's MIC field and generate the appropriate MIC status.

Decryption is started by triggering the CRYPT task with the MODE on page 111 register set to DECRYPTION. An ENDCRYPT event will be generated when packet decryption is completed

The AES CCM will also modify the length field of the packet to adjust for the MIC field, that is, subtract four bytes from the length, and then store the decrypted packet into RAM at the address pointed to by the OUTPTR pointer, see Decryption on page 106.

The CCM is only able to decrypt packet payloads that are at least 5 bytes long, that is, 1 byte or more encrypted payload (EPL) and 4 bytes of MIC. The CCM will therefore generate a MIC error for packets where the length field is set to 1, 2, 3 or 4.

Empty packets (length field is set to 0) will not be decrypted but instead moved unmodified through the AES CCM, these packets will always pass the MIC check.

The CCM supports different widths of the LENGTH field in the data structure for decrypted packets. This is configured in the MODE on page 111 register.

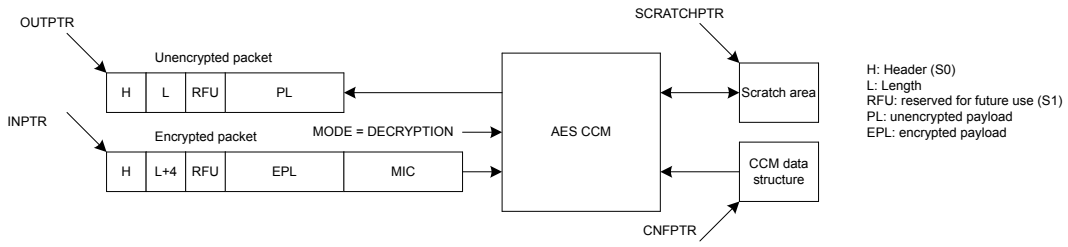

Figure 31: Decryption

6.4.4 AES CCM and RADIO concurrent operation

The CCM module is able to encrypt/decrypt data synchronously to data being transmitted or received on the radio.

In order for the CCM module to run synchronously with the radio, the data rate setting in the MODE on page 111 register needs to match the radio data rate. The settings in this register apply whenever either the KSGEN or CRYPT tasks are triggered.

The data rate setting of the MODE on page 111 register can also be overridden on-the-fly during an ongoing encrypt/decrypt operation by the contents of the RATEOVERRIDE on page 113 register. The data rate setting in this register applies whenever the RATEOVERRIDE task is triggered. This feature can be useful in cases where the radio data rate is changed during an ongoing packet transaction.

6.4.5 Encrypting packets on-the-fly in radio transmit mode

When the AES CCM is encrypting a packet on-the-fly at the same time as the radio is transmitting it, the radio must read the encrypted packet from the same memory location as the AES CCM is writing to.

The OUTPTR on page 112 pointer in the AES CCM must therefore point to the same memory location as the PACKETPTR pointer in the radio, see Configuration of on-the-fly encryption on page 106.

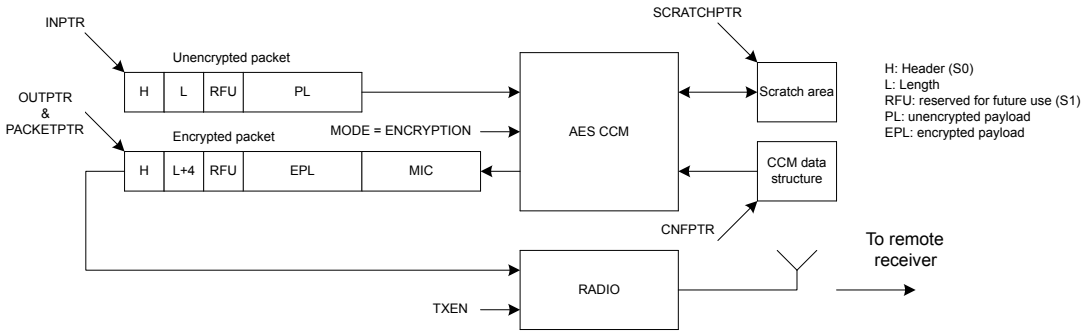

Figure 32: Configuration of on-the-fly encryption

In order to match the RADIO's timing, the KSGEN task must be triggered early enough to allow the key-stream generation to complete before the encryption of the packet shall start.

For short packets (MODE.LENGTH = Default) the KSGEN task must be triggered no later than when the START task in the RADIO is triggered. In addition the shortcut between the ENDKSGEN event and the CRYPT task must be enabled. This use-case is illustrated in On-the-fly encryption of short packets (MODE.LENGTH = Default) using a PPI connection on page 107 using a PPI connection between the READY event in the RADIO and the KSGEN task in the AES CCM.

For long packets (MODE.LENGTH = Extended) the key-stream generation will need to be started even earlier, for example at the time when the TXEN task in the RADIO is triggered.

Important: Refer to Timing specification on page 114 for information about the time needed for generating a key-stream.

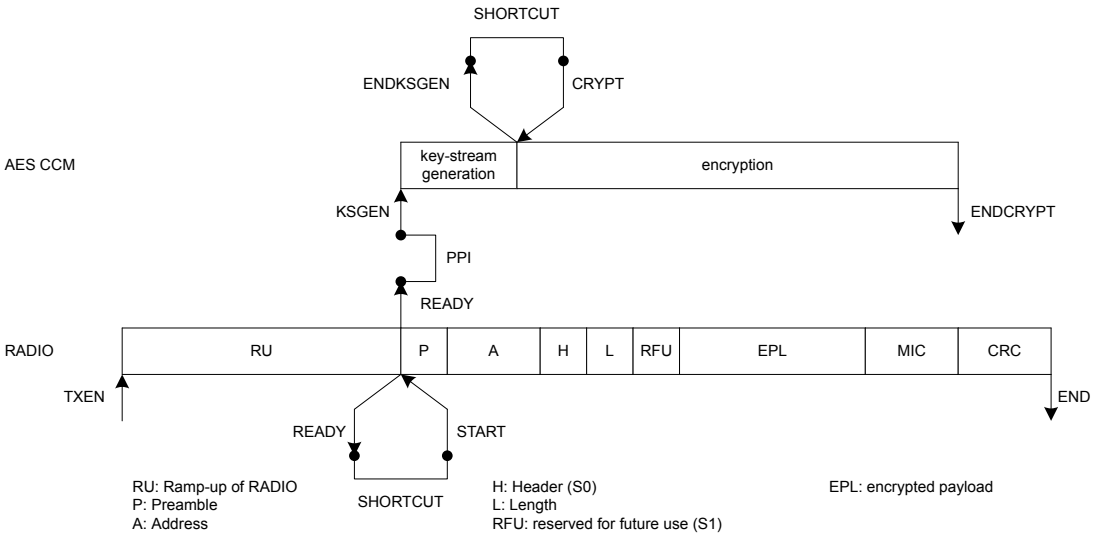

Figure 33: On-the-fly encryption of short packets (MODE.LENGTH = Default) using a PPI connection

6.4.6 Decrypting packets on-the-fly in radio receive mode

When the AES CCM is decrypting a packet on-the-fly at the same time as the RADIO is receiving it, the AES CCM must read the encrypted packet from the same memory location as the RADIO is writing to.

The INPTR on page 112 pointer in the AES CCM must therefore point to the same memory location as the PACKETPTR pointer in the RADIO, see Configuration of on-the-fly decryption on page 107.

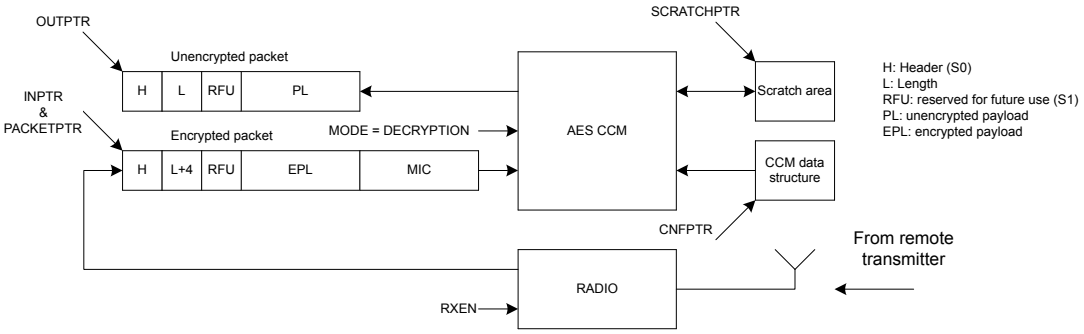

Figure 34: Configuration of on-the-fly decryption

In order to match the RADIO's timing, the KSGEN task must be triggered early enough to allow the key-stream generation to complete before the decryption of the packet shall start.

For short packets (MODE.LENGTH = Default) the KSGEN task must be triggered no later than when the START task in the RADIO is triggered. In addition, the CRYPT task must be triggered no earlier than when the ADDRESS event is generated by the RADIO.

If the CRYPT task is triggered exactly at the same time as the ADDRESS event is generated by the RADIO, the AES CCM will guarantee that the decryption is completed no later than when the END event in the RADIO is generated.

This use-case is illustrated in On-the-fly decryption of short packets (MODE.LENGTH = Default) using a PPI connection on page 108 using a PPI connection between the ADDRESS event in the RADIO and the CRYPT task in the AES CCM. The KSGEN task is triggered from the READY event in the RADIO through a PPI connection.

For long packets (MODE.LENGTH = Extended) the key-stream generation will need to be started even earlier, for example at the time when the RXEN task in the RADIO is triggered.

Important: Refer to Timing specification on page 114 for information about the time needed for generating a key-stream.

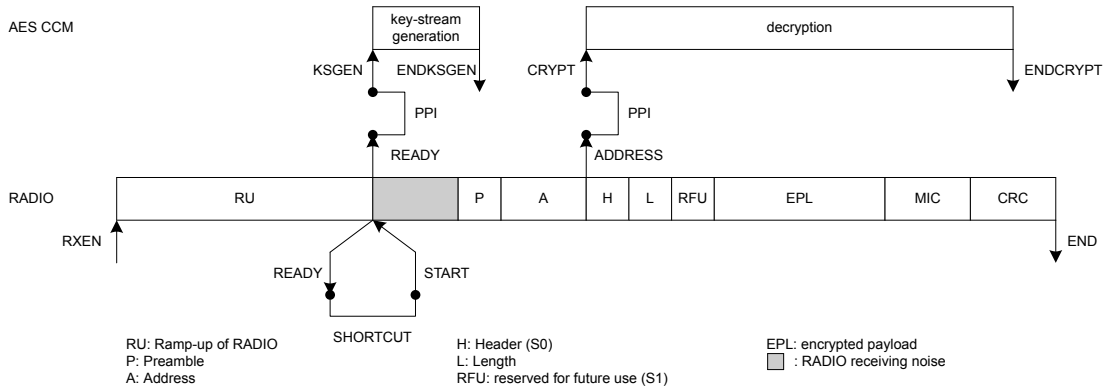

Figure 35: On-the-fly decryption of short packets (MODE.LENGTH = Default) using a PPI connection

6.4.7 CCM data structure

The CCM data structure is located in Data RAM at the memory location specified by the CNFPTR pointer register.

Property	Address offset	Description
KEY	0	16 byte AES key
PKTCTR	16	Octet0 (LSO) of packet counter
	17	Octet1 of packet counter
	18	Octet2 of packet counter
	19	Octet3 of packet counter
	20	Bit 6 – Bit 0: Octet4 (7 most significant bits of packet counter, with Bit 6 being the most significant bit) Bit7: Ignored
	21	Ignored
	22	Ignored
	23	Ignored
	24	Bit 0: Direction bit Bit 7 – Bit 1: Zero padded
IV	25	8 byte initialization vector (IV) Octet0 (LSO) of IV, Octet1 of IV, … , Octet7 (MSO) of IV

Table 26: CCM data structure overview

The NONCE vector (as specified by the *Bluetooth* Core Specification) will be generated by hardware based on the information specified in the CCM data structure from CCM data structure overview on page 108 .

Property	Address offset	Description
HEADER	0	Packet Header
LENGTH	1	Number of bytes in unencrypted payload
RFU	2	Reserved Future Use
PAYLOAD	3	Unencrypted payload

Table 27: Data structure for unencrypted packet

Property	Address offset	Description
HEADER	0	Packet Header
LENGTH	1	Number of bytes in encrypted payload including length of MIC **Important:** LENGTH will be 0 for empty packets since the MIC is not added to empty packets
RFU	2	Reserved Future Use
PAYLOAD	3	Encrypted payload
MIC	3 + payload length	ENCRYPT: 4 bytes encrypted MIC **Important:** MIC is not added to empty packets

Table 28: Data structure for encrypted packet

6.4.8 EasyDMA and ERROR event

The CCM implements an EasyDMA mechanism for reading and writing to the RAM.

In cases where the CPU and other EasyDMA enabled peripherals are accessing the same RAM block at the same time, a high level of bus collisions may cause too slow operation for correct on the fly encryption. In this case the ERROR event will be generated.

The EasyDMA will have finished accessing the RAM when the ENDKSGEN and ENDCRYPT events are generated.

If the CNFPTR, SCRATCHPTR, INPTR and the OUTPTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

6.4.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption	

Table 29: Instances

Register	Offset	Description	
TASKS_KSGEN	0x000	Start generation of key-stream. This operation will stop by itself when completed.	
TASKS_CRYPT	0x004	Start encryption/decryption. This operation will stop by itself when completed.	
TASKS_STOP	0x008	Stop encryption/decryption	
TASKS_RATEOVERRIDE	0x00C	Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption	
EVENTS_ENDKSGEN	0x100	Key-stream generation complete	
EVENTS_ENDCRYPT	0x104	Encrypt/decrypt complete	
EVENTS_ERROR	0x108	CCM error event	Deprecated
SHORTS	0x200	Shortcut register	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
MICSTATUS	0x400	MIC check result	
ENABLE	0x500	Enable	
MODE	0x504	Operation mode	
CNFPTR	0x508	Pointer to data structure holding AES key and NONCE vector	
INPTR	0x50C	Input pointer	
OUTPTR	0x510	Output pointer	

Register	Offset	Description
SCRATCHPTR	0x514	Pointer to data area used for temporary storage
MAXPACKETSIZE	0x518	Length of key-stream generated when MODE.LENGTH = Extended.
RATEOVERRIDE	0x51C	Data rate override setting.

Table 30: Register overview

6.4.9.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENDKSGEN_CRYPT			Shortcut between ENDKSGEN event and CRYPT task See EVENTS_ENDKSGEN and TASKS_CRYPT
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.4.9.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENDKSGEN			Write '1' to enable interrupt for ENDKSGEN event See EVENTS_ENDKSGEN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDCRYPT			Write '1' to enable interrupt for ENDCRYPT event See EVENTS_ENDCRYPT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.4.9.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENDKSGEN			Write '1' to disable interrupt for ENDKSGEN event See EVENTS_ENDKSGEN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDCRYPT			Write '1' to disable interrupt for ENDCRYPT event See EVENTS_ENDCRYPT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.4.9.4 MICSTATUS

Address offset: 0x400

MIC check result

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	MICSTATUS			The result of the MIC check performed during the previous decryption operation
			CheckFailed	0	MIC check failed
			CheckPassed	1	MIC check passed

6.4.9.5 ENABLE

Address offset: 0x500

Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable CCM
			Disabled	0	Disable
			Enabled	2	Enable

6.4.9.6 MODE

Address offset: 0x504

Operation mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	MODE			The mode of operation to be used. The settings in this register apply whenever either the KSGEN or CRYPT tasks are triggered.
			Encryption	0	AES CCM packet encryption mode
			Decryption	1	AES CCM packet decryption mode
B	RW	DATARATE			Radio data rate that the CCM shall run synchronous with
			1Mbit	0	1 Mbps
			2Mbit	1	2 Mbps
			125Kbps	2	125 Kbps
			500Kbps	3	500 Kbps
C	RW	LENGTH			Packet length configuration
			Default	0	Default length. Effective length of LENGTH field in encrypted/decrypted packet is 5 bits. A key-stream for packet payloads up to 27 bytes will be generated.
			Extended	1	Extended length. Effective length of LENGTH field in encrypted/decrypted packet is 8 bits. A key-stream for packet payloads up to MAXPACKETSIZE bytes will be generated.

6.4.9.7 CNFPTR

Address offset: 0x508

Pointer to data structure holding AES key and NONCE vector

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CNFPTR			Pointer to the data structure holding the AES key and the CCM NONCE vector (see Table 1 CCM data structure overview)

6.4.9.8 INPTR

Address offset: 0x50C

Input pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	INPTR			Input pointer

6.4.9.9 OUTPTR

Address offset: 0x510

Output pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	OUTPTR			Output pointer

6.4.9.10 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SCRATCHPTR			Pointer to a scratch data area used for temporary storage during key-stream generation, MIC generation and encryption/decryption. The scratch area is used for temporary storage of data during key-stream generation and encryption. When MODE.LENGTH = Default, a space of 43 bytes is required for this temporary storage. MODE.LENGTH = Extended (16 + MAXPACKETSIZE) bytes of storage is required.

6.4.9.11 MAXPACKETSIZE

Address offset: 0x518

Length of key-stream generated when MODE.LENGTH = Extended.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x000000FB	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXPACKETSIZE		[0x001B..0x00FB]	Length of key-stream generated when MODE.LENGTH = Extended. This value must be greater or equal to the subsequent packet payload to be encrypted/decrypted.

6.4.9.12 RATEOVERRIDE

Address offset: 0x51C

Data rate override setting.

Override value to be used instead of the setting of MODE.DATARATE. This override value applies when the RATEOVERRIDE task is triggered.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value																															
A	RW	RATEOVERRIDE																																	
			1Mbit	0																															
			2Mbit	1																															
			125Kbps	2																															
			500Kbps	3																															

6.4.10 Electrical specification

6.4.10.1 Timing specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{kgen}	Time needed for key-stream generation (given priority access to destination RAM block).			50	µs

6.5 COMP — Comparator

The comparator (COMP) compares an input voltage (VIN+) against a second input voltage (VIN-). VIN+ can be derived from an analog input pin (AIN0-AIN7). VIN- can be derived from multiple sources depending on the operation mode of the comparator.

Main features of the comparator are:

- Input range from 0 V to VDD
- Single-ended mode
  - Fully flexible hysteresis using a 64-level reference ladder
- Differential mode
  - Configurable 50 mV hysteresis
- Reference inputs (VREF):
  - VDD
  - External reference from AIN0 to AIN7 (between 0 V and VDD)
  - Internal references 1.2 V, 1.8 V and 2.4 V
- Three speed/power consumption modes: low-power, normal and high-speed
- Single-pin capacitive sensor support
- Event generation on output changes
  - UP event on VIN- > VIN+
  - DOWN event on VIN- < VIN+
  - CROSS event on VIN+ and VIN- crossing
  - READY event on core and internal reference (if used) ready

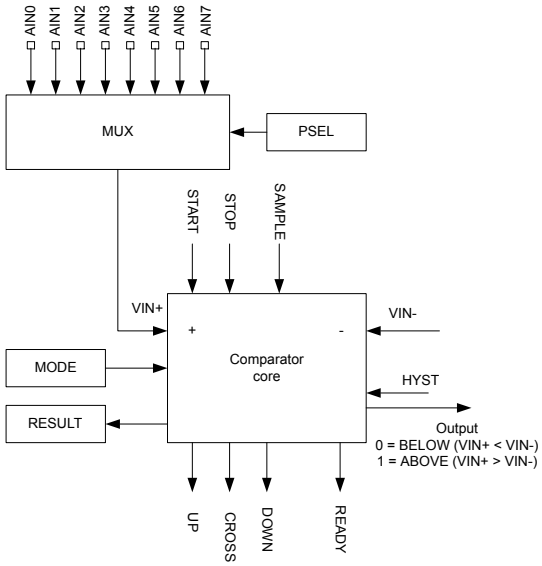

Figure 36: Comparator overview

Once enabled (using the ENABLE register), the comparator is started by triggering the START task and stopped by triggering the STOP task. After a start-up time of $t_{COMP,START}$, the comparator will generate a READY event to indicate that it is ready for use and that its output is correct. When the COMP module is started, events will be generated every time VIN+ crosses VIN-.

Operation modes

The comparator can be configured to operate in two main operation modes, differential mode and single-ended mode. See the MODE register for more information. In both operation modes, the comparator can operate in different speed and power consumption modes (low-power, normal and high-speed). High-speed mode will consume more power compared to low-power mode, and low-power mode will result in slower response time compared to high-speed mode.

Use the PSEL register to select any of the AIN0-AIN7 pins as VIN+ input, irregardless of the operation mode selected for the comparator. The source of VIN- depends on which operation mode is used:

- Differential mode: Derived directly from AIN0 to AIN7
- Single-ended mode: Derived from VREF. VREF can be derived from VDD, AIN0-AIN7 or internal 1.2 V, 1.8 V and 2.4 V references.

The selected analog pins will be acquired by the comparator once it is enabled.

An optional hysteresis on VIN+ and VIN- can be enabled when the module is used in differential mode through the HYST register. In single-ended mode, VUP and VDOWN thresholds can be set to implement a hysteresis using the reference ladder (see Comparator in single-ended mode on page 117). This hysteresis is in the order of magnitude of 50 mV, and shall prevent noise on the signal to create unwanted events. See Hysteresis example where VIN+ starts below VUP on page 118 for illustration of the effect of an active hysteresis on a noisy input signal.

An upward crossing will generate an UP event and a downward crossing will generate a DOWN event. The CROSS event will be generated every time there is a crossing, independent of direction.

The immediate value of the comparator can be sampled to RESULT register by triggering the SAMPLE task.

6.5.1 Differential mode

In differential mode, the reference input VIN- is derived directly from one of the AINx pins.

Before enabling the comparator via the ENABLE register, the following registers must be configured for the differential mode:

- PSEL
- MODE
- EXTREFSEL

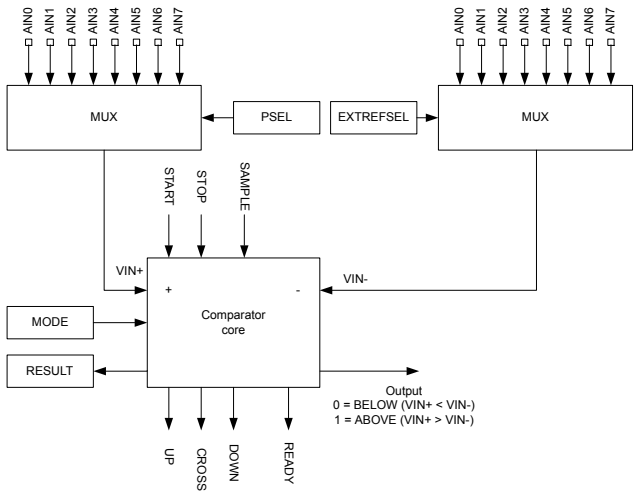

Figure 37: Comparator in differential mode

Restriction: Depending on the device, not all the analog inputs may be available for each MUX. See definitions for PSEL and EXTREFSEL for more information about which analog pins are available on a particular device.

When HYST register is turned on while in this mode, the output of the comparator (and associated events) will change from ABOVE to BELOW whenever VIN+ becomes lower than VIN- - ($V_{DIFFHYST}$ / 2). It will also change from BELOW to ABOVE whenever VIN+ becomes higher than VIN- + ($V_{DIFFHYST}$ / 2). This behavior is illustrated in Hysteresis enabled in differential mode on page 116.

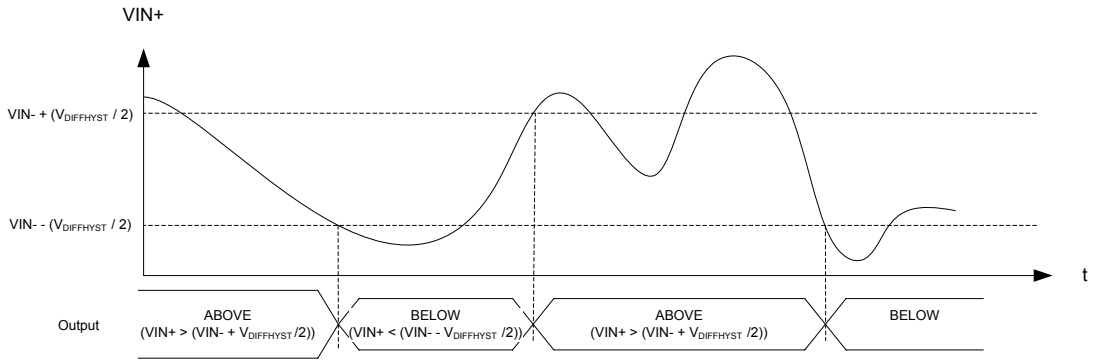

Figure 38: Hysteresis enabled in differential mode

6.5.2 Single-ended mode

In single-ended mode, VIN- is derived from the reference ladder.

Before enabling the comparator via the ENABLE register, the following registers must be configured for the single-ended mode:

- PSEL

- MODE
- REFSEL
- EXTREFSEL
- TH

The reference ladder uses the reference voltage (VREF) to derive two new voltage references, VUP and VDOWN. VUP and VDOWN are configured using THUP and THDOWN respectively in the TH register. VREF can be derived from any of the available reference sources, configured using the EXTREFSEL and REFSEL registers as illustrated in Comparator in single-ended mode on page 117. When AREF is selected in the REFSEL register, the EXTREFSEL register is used to select one of the AIN0-AIN7 analog input pins as reference input. The selected analog pins will be acquired by the comparator once it is enabled.

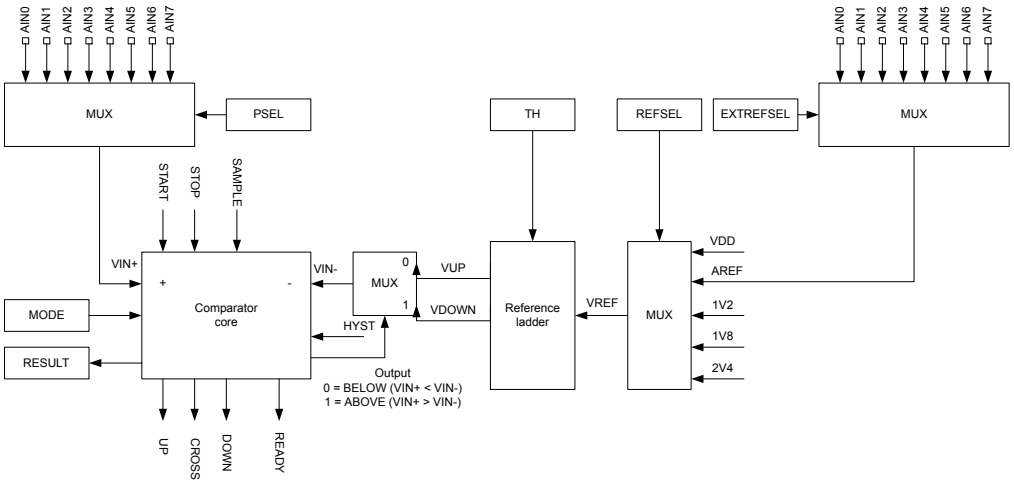

Figure 39: Comparator in single-ended mode

Restriction: Depending on the device, not all the analog inputs may be available for each MUX. See definitions for PSEL and EXTREFSEL for more information about which analog pins are available on a particular device.

When the comparator core detects that VIN+ > VIN-, i.e. ABOVE as per the RESULT register, VIN- will switch to VDOWN. When VIN+ falls below VIN- again, VIN- will be switched back to VUP. By specifying VUP larger than VDOWN, a hysteresis can be generated as illustrated in Hysteresis example where VIN+ starts below VUP on page 118 and Hysteresis example where VIN+ starts above VUP on page 118.

Writing to HYST has no effect in single-ended mode, and the content of this register is ignored.

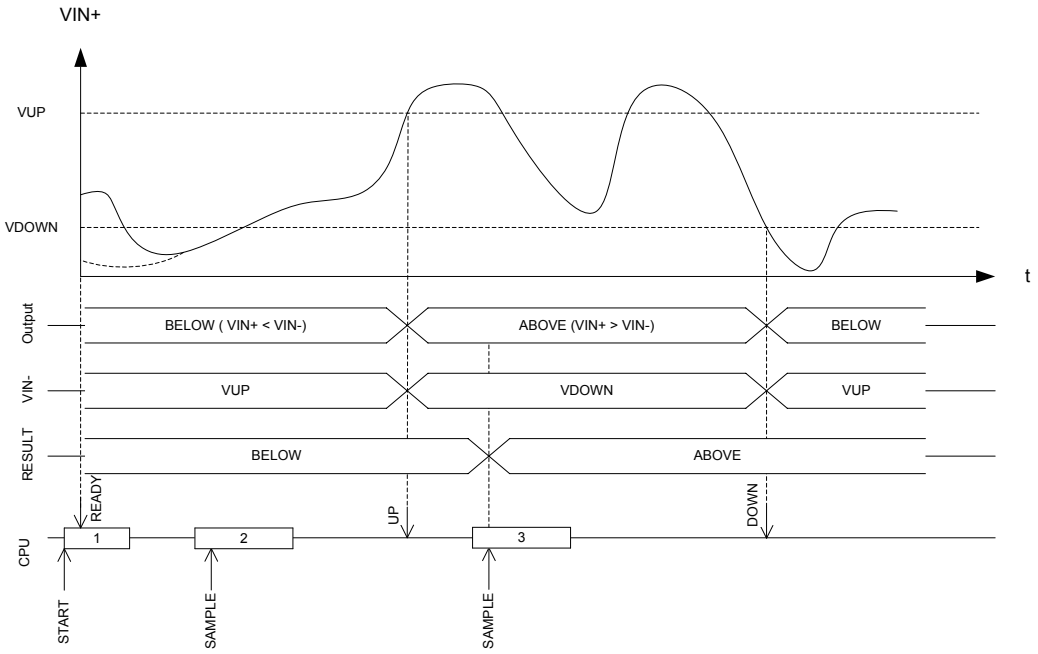

Figure 40: Hysteresis example where VIN+ starts below VUP

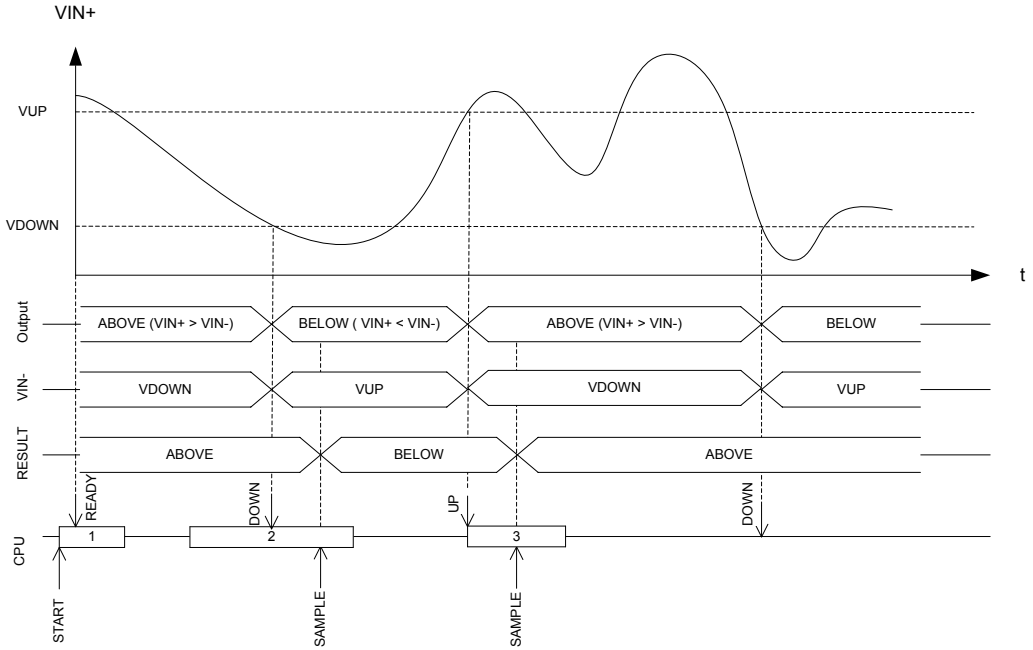

Figure 41: Hysteresis example where VIN+ starts above VUP

6.5.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40013000	COMP	COMP	General purpose comparator	

Table 31: Instances

Register	Offset	Description
TASKS_START	0x000	Start comparator

Register	Offset	Description
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	COMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RESULT	0x400	Compare result
ENABLE	0x500	COMP enable
PSEL	0x504	Pin select
REFSEL	0x508	Reference source select for single-ended mode
EXTREFSEL	0x50C	External reference select
TH	0x530	Threshold configuration for hysteresis unit
MODE	0x534	Mode configuration
HYST	0x538	Comparator hysteresis enable

Table 32: Register overview

6.5.3.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value	Description																														
A	RW	READY_SAMPLE			Shortcut between READY event and SAMPLE task See EVENTS_READY and TASKS_SAMPLE																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
B	RW	READY_STOP			Shortcut between READY event and STOP task See EVENTS_READY and TASKS_STOP																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
C	RW	DOWN_STOP			Shortcut between DOWN event and STOP task See EVENTS_DOWN and TASKS_STOP																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
D	RW	UP_STOP			Shortcut between UP event and STOP task See EVENTS_UP and TASKS_STOP																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
E	RW	CROSS_STOP			Shortcut between CROSS event and STOP task See EVENTS_CROSS and TASKS_STOP																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																

6.5.3.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Enable or disable interrupt for READY event See EVENTS_READY
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	DOWN			Enable or disable interrupt for DOWN event See EVENTS_DOWN
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	UP			Enable or disable interrupt for UP event See EVENTS_UP
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	CROSS			Enable or disable interrupt for CROSS event See EVENTS_CROSS
			Disabled	0	Disable
			Enabled	1	Enable

6.5.3.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to enable interrupt for DOWN event See EVENTS_DOWN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to enable interrupt for UP event See EVENTS_UP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
D	RW	CROSS			Write '1' to enable interrupt for CROSS event See EVENTS_CROSS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.5.3.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to disable interrupt for DOWN event See EVENTS_DOWN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to disable interrupt for UP event See EVENTS_UP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to disable interrupt for CROSS event See EVENTS_CROSS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.5.3.5 RESULT

Address offset: 0x400

Compare result

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RESULT			Result of last compare. Decision point SAMPLE task.
			Below	0	Input voltage is below the threshold (VIN+ < VIN-)
			Above	1	Input voltage is above the threshold (VIN+ > VIN-)

6.5.3.6 ENABLE

Address offset: 0x500

COMP enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable COMP
			Disabled	0	Disable
			Enabled	2	Enable

6.5.3.7 PSEL

Address offset: 0x504

Pin select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PSEL			Analog pin select
			AnalogInput0	0	AIN0 selected as analog input
			AnalogInput1	1	AIN1 selected as analog input
			AnalogInput2	2	AIN2 selected as analog input
			AnalogInput3	3	AIN3 selected as analog input
			AnalogInput4	4	AIN4 selected as analog input
			AnalogInput5	5	AIN5 selected as analog input
			AnalogInput6	6	AIN6 selected as analog input
			AnalogInput7	7	AIN7 selected as analog input

6.5.3.8 REFSEL

Address offset: 0x508

Reference source select for single-ended mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REFSEL			Reference select
			Int1V2	0	VREF = internal 1.2 V reference (VDD >= 1.7 V)
			Int1V8	1	VREF = internal 1.8 V reference (VDD >= VREF + 0.2 V)
			Int2V4	2	VREF = internal 2.4 V reference (VDD >= VREF + 0.2 V)
			VDD	4	VREF = VDD
			ARef	5	VREF = AREF (VDD >= VREF >= AREFMIN)

6.5.3.9 EXTREFSEL

Address offset: 0x50C

External reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EXTREFSEL			External analog reference select
			AnalogReference0	0	Use AIN0 as external analog reference
			AnalogReference1	1	Use AIN1 as external analog reference
			AnalogReference2	2	Use AIN2 as external analog reference
			AnalogReference3	3	Use AIN3 as external analog reference
			AnalogReference4	4	Use AIN4 as external analog reference
			AnalogReference5	5	Use AIN5 as external analog reference
			AnalogReference6	6	Use AIN6 as external analog reference
			AnalogReference7	7	Use AIN7 as external analog reference

6.5.3.10 TH

Address offset: 0x530

Threshold configuration for hysteresis unit

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	THDOWN		[63:0]	VDOWN = (THDOWN+1)/64*VREF
B	RW	THUP		[63:0]	VUP = (THUP+1)/64*VREF

6.5.3.11 MODE

Address offset: 0x534

Mode configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SP			Speed and power modes
			Low	0	Low-power mode
			Normal	1	Normal mode
			High	2	High-speed mode
B	RW	MAIN			Main operation modes
			SE	0	Single-ended mode
			Diff	1	Differential mode

6.5.3.12 HYST

Address offset: 0x538

Comparator hysteresis enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	HYST			Comparator hysteresis
			NoHyst	0	Comparator hysteresis disabled
			Hyst50mV	1	Comparator hysteresis enabled

6.5.4 Electrical specification

6.5.4.1 COMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{PROPDLY,LP}$	Propagation delay, low-power mode[a]		0.6		µS
$t_{PROPDLY,N}$	Propagation delay, normal mode[a]		0.2		µS
$t_{PROPDLY,HS}$	Propagation delay, high-speed mode[a]		0.1		µS
$V_{DIFFHYST}$	Optional hysteresis applied to differential input		30		mV
$V_{VDD-VREF}$	Required difference between VDD and a selected VREF, VDD > VREF	0.3			V
$t_{INT\_REF,START}$	Startup time for the internal bandgap reference		50	80	µS
E_{INT_REF}	Internal bandgap reference error	-3		3	%
$V_{INPUTOFFSET}$	Input offset	-10		10	mV
$t_{COMP,START}$	Startup time for the comparator core		3		µS

Total comparator run current must be calculated from the I_{COMP}, I_{INT_REF}, and I_{LADDER} values for a given reference voltage.

6.6 CRYPTOCELL — ARM TrustZone CryptoCell 310

ARM® TrustZone® CryptoCell 310 (CRYPTOCELL) is a security subsystem which provides root of trust (RoT) and cryptographic services for a device.

[a] Propagation delay is with 10 mV overdrive.

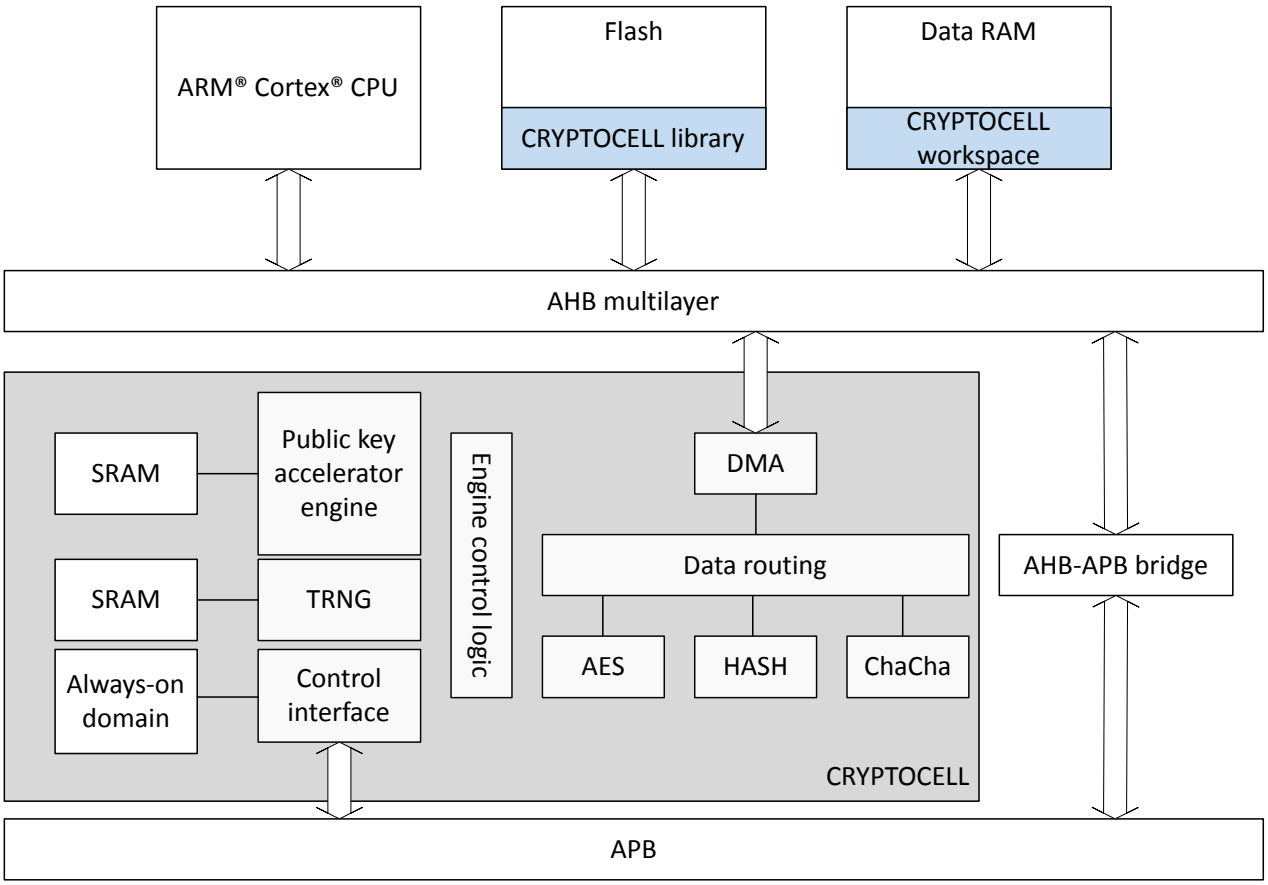

Figure 42: Block diagram for CRYPTOCELL

The following cryptographic features are provided:

- True random number generator (TRNG) compliant with NIST 800-90B[16], AIS-31, and FIPS 140-2/3[16].
- Pseudorandom number generator (PRNG) using underlying AES engine compliant with NIST 800-90A
- RSA public key cryptography
  - Up to 2048-bit key size
  - PKCS#1 v2.1/v1.5
  - Optional CRT support
- Elliptic curve cryptography (ECC)
  - NIST FIPS 186-4 recommended curves using pseudorandom parameters, up to 521 bits:
    - Prime field: P-192, P-224, P-256, P-384, P-521
  - SEC 2 recommended curves using pseudorandom parameters, up to 521 bits:
    - Prime field: secp160r1, secp192r1, secp224r1, secp256r1, secp384r1, secp521r1
  - Koblitz curves using fixed parameters, up to 256 bits:
    - Prime field: secp160k1, secp192k1, secp224k1, secp256k1
  - Edwards/Montgomery curves:
    - Ed25519, Curve25519
  - ECDH/ECDSA support
- Secure remote password protocol (SRP)
  - Up to 3072-bit operations

[16] Not finalized at time of publishing (draft)

- Hashing functions
  - SHA-1, SHA-2 up to 256 bits
  - Keyed-hash message authentication code (HMAC)
- AES symmetric encryption
  - General purpose AES engine (encrypt/decrypt, sign/verify)
  - 128-bit key size
  - Supported encryption modes: ECB, CBC, CMAC/CBC-MAC, CTR, CCM/CCM*
- ChaCha20/Poly1305 symmetric encryption
  - Supported key size: 128 and 256 bits
  - Authenticated encryption with associated data (AEAD) mode

6.6.1 Usage

The CRYPTOCELL state is controlled via a register interface. The cryptographic functions of CRYPTOCELL are accessible by using a software library provided in the device SDK, not directly via a register interface.

To enable CRYPTOCELL, use register ENABLE on page 129.

6.6.2 Always-on (AO) power domain

The CRYPTOCELL subsystem has an internal always-on (AO) power domain for retaining device secrets when CRYPTOCELL is disabled.

The following information is retained by the AO power domain:

- 4 bits indicating the configured CRYPTOCELL life-cycle state (LCS)
- 1 bit indicating if RTL key K_{PRTL} is available for use
- 128-bit device root key K_{DR}

A reset from any reset source will erase the content in the AO power domain.

6.6.3 Lifecycle state (LCS)

Lifecycle refers to multiple states a device goes through during its lifetime. Two valid lifecycle states are offered for the device - debug and secure.

The CRYPTOCELL subsystem lifecycle state (LCS) is controlled through register HOST_IOT_LCS on page 131. A valid LCS is configured by writing either value `Debug` or `Secure` into the LCS field of this register. A correctly configured LCS can be validated by reading back the read-only field LCS_IS_VALID from the abovementioned register. The LCS_IS_VALID field value will change from `Invalid` to `Valid` once a valid LCS value has been written.

LCS field value	LCS_IS_VALID field value	Description
Secure	Invalid	Default reset value indicating that LCS has not been configured.
Secure	Valid	LCS set to secure mode, and LCS is valid. Registers HOST_IOT_KDR[0..3] can only be written once per reset cycle. Any additional writes will be ignored.
Debug	Valid	LCS set to debug mode, and LCS is valid. Registers HOST_IOT_KDR[0..3] can be written multiple times.

Table 33: Lifecycle states

6.6.4 Cryptographic key selection

The CRYPTOCELL subsystem can be instructed to operate on different cryptographic keys.

Through register HOST_CRYPTOKEY_SEL on page 129, the following key types can be selected for cryptographic operations:

- RTL key K_{PRTL}
- Device root key K_{DR}
- Session key

K_{PRTL} and K_{DR} are configured as part of the CRYPTOCELL initialization process, while session keys are provided by the application through the software library API.

6.6.4.1 RTL key

The ARM® TrustZone® CryptoCell 310 IP contains one hard-coded RTL key referred to as K_{PRTL}. This key is set to the same value for all devices with the same part code in the hardware design and cannot be changed.

The K_{PRTL} key can be requested for use in cryptographic operations by the CRYPTOCELL, without revealing the key value itself. Access to use of K_{PRTL} in cryptographic operations can be disabled until next reset by writing to register HOST_IOT_KPRTL_LOCK on page 130. If a locked K_{PRTL} key is requested for use, a zero vector key will be routed to the AES engine instead.

6.6.4.2 Device root key

The device root key K_{DR} is a 128-bit AES key programmed into the CRYPTOCELL subsystem using firmware. It is retained in the AO power domain until the next reset.

Once configured, it is possible to perform cryptographic operations using the the CRYPTOCELL subsystem where K_{DR} is selected as key input without having access to the key value itself. The K_{DR} key value must be written to registers HOST_IOT_KDR[0..3]. These 4 registers are write-only if LCS is set to debug mode, and write-once if LCS is set to secure mode. The K_{DR} key value is successfully retained when the read-back value of register HOST_IOT_KDR0 on page 130 changes to `1`.

6.6.5 Direct memory access (DMA)

The CRYPTOCELL subsystem implements direct memory access (DMA) for accessing memory without CPU intervention.

The following table shows which memory type(s) can be accessed using the DMA:

SRAM		Flash		External flash (QSPI)	
Read	**Write**	**Read**	**Write**	**Read**	**Write**
Yes	Yes	No	No	No	No

Table 34: DMA transaction types

Any data stored in memory type(s) not accessible by the DMA engine must be copied to SRAM before it can be processed by the CRYPTOCELL subsystem. Maximum DMA transaction size is limited to 2^{16}-1 bytes.

6.6.6 Standards

ARM® TrustZone® CryptoCell 310 (CRYPTOCELL) supports a number of cryptography standards.

Algorithm family	Identification code	Document title
TRNG	NIST 800-90B[17]	*Recommendation for the Entropy Sources Used for Random Bit Generation*
	AIS-31	*A proposal for: Functionality classes and evaluation methodology for physical random number generators*
	FIPS 140-2/3[17]	*Security Requirements for Cryptographic Modules*
PRNG	NIST 800-90A	*Recommendation for Random Number Generation Using Deterministic Random Bit Generators*
Stream cipher	Chacha	*ChaCha, a variant of Salsa20*, Daniel J. Bernstein, January 28th 2008
MAC	Poly1305	*The Poly1305-AES message-authentication code*, Daniel J. Bernstein *Cryptography in NaCl*, Daniel J. Bernstein
Key agreement	SRP	*The Secure Remote Password Protocol*, Thomas Wu, November 11th 1997
AES	FIPS-197	*Advanced Encryption Standard (AES)*
	NIST SP 800-38A	*Recommendation for Block Cipher Modes of Operation - Methods and Techniques*
	NIST SP 800-38B	*Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*
	NIST SP 800-38C	*Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*
	ISO/IEC 9797-1	AES CBC-MAC per ISO/IEC 9797-1 MAC algorithm 1
	IEEE 802.15.4-2011	*IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Annex B.4: *Specification of generic CCM* mode of operation*
HMAC	RFC2104	*HMAC: Keyed-Hashing for Message Authentication*
RSA	PKCS#1	*Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications* v1.5/2.1
Diffie-Hellman	ANSI X9.42	*Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*
	PKCS#3	*Diffie-Hellman Key-Agreement Standard*
ECC	ANSI X9.63	*Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography*
	IEEE 1363	*Standard Specifications for Public-Key Cryptography*
	ANSI X9.62	*Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*
	Ed25519	Edwards-curve, *Ed25519: high-speed high-security signatures*, Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang
	Curve25519	Montgomery curve, *Curve25519: new Diffie-Hellman speed records*, Daniel J. Bernstein
	FIPS 186-4	*Digital Signature Standard (DSS)*
	SEC 2	*Recommended Elliptic Curve Domain Parameters*, Certicom Research
	NIST SP 800-56A rev. 2	*Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*
General	FIPS 140-2	*Security Requirements for Cryptographic Modules*

Table 35: CRYPTOCELL cryptography standards

6.6.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x5002A000	CRYPTOCELL	CRYPTOCELL	CryptoCell subsystem control interface	

Table 36: Instances

Register	Offset	Description
ENABLE	0x500	Enable CRYPTOCELL subsystem

Table 37: Register overview

[17] Not finalized at time of publishing (draft)

6.6.7.1 ENABLE

Address offset: 0x500

Enable CRYPTOCELL subsystem

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable the CRYPTOCELL subsystem
			Disabled	0	CRYPTOCELL subsystem disabled
			Enabled	1	CRYPTOCELL subsystem enabled
					When enabled the CRYPTOCELL subsystem can be initialized and controlled through the CryptoCell firmware API

6.6.8 Host interface

This chapter describe host registers used for controlling the CRYPTOCELL subsystem behavior.

6.6.8.1 HOST_RGF block

The HOST_RGF block contains registers for configuring LCS and device root key K_{DR}, in addition to selecting which cryptographic key is connected to the AES engine.

6.6.8.1.1 Registers

Register	Offset	Description
HOST_CRYPTOKEY_SEL	0x1A38	AES hardware key select
HOST_IOT_KPRTL_LOCK	0x1A4C	This write-once register is the K_PRTL lock register. When this register is set, K_PRTL can not be used and a zeroed key will be used instead. The value of this register is saved in the CRYPTOCELL AO power domain.
HOST_IOT_KDR0	0x1A50	This register holds bits 31:0 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain. Reading from this address returns the K_DR valid status indicating if K_DR is successfully retained.
HOST_IOT_KDR1	0x1A54	This register holds bits 63:32 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.
HOST_IOT_KDR2	0x1A58	This register holds bits 95:64 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.
HOST_IOT_KDR3	0x1A5C	This register holds bits 127:96 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.
HOST_IOT_LCS	0x1A60	Controls lifecycle state (LCS) for CRYPTOCELL subsystem

Table 38: Register overview

6.6.8.1.1.1 HOST_CRYPTOKEY_SEL

Address offset: 0x1A38

AES hardware key select

If the HOST_IOT_KPRTL_LOCK register is set, and the HOST_CRYPTOKEY_SEL register set to 1, then the HW key that is connected to the AES engine is zero

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	HOST_CRYPTOKEY_SEL			Select the source of the HW key that is used by the AES engine
			K_DR	0	Use device root key K_DR from CRYPTOCELL AO power domain
			K_PRTL	1	Use hard-coded RTL key K_PRTL
			Session	2	Use provided session key

6.6.8.1.1.2 HOST_IOT_KPRTL_LOCK

Address offset: 0x1A4C

This write-once register is the K_PRTL lock register. When this register is set, K_PRTL can not be used and a zeroed key will be used instead. The value of this register is saved in the CRYPTOCELL AO power domain.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	HOST_IOT_KPRTL_LOCK			This register is the K_PRTL lock register. When this register is set, K_PRTL can not be used and a zeroed key will be used instead. The value of this register is saved in the CRYPTOCELL AO power domain.
			Disabled	0	K_PRTL can be selected for use from register HOST_CRYPTOKEY_SEL
			Enabled	1	K_PRTL has been locked until next power-on reset (POR). If K_PRTL is selected anyway, a zeroed key will be used instead.

6.6.8.1.1.3 HOST_IOT_KDR0

Address offset: 0x1A50

This register holds bits 31:0 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain. Reading from this address returns the K_DR valid status indicating if K_DR is successfully retained.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	HOST_IOT_KDR0			Write: K_DR bits 31:0 Read: 0x00000000 when 128-bit K_DR key value is not yet retained in the CRYPTOCELL AO power domain Read: 0x00000001 when 128-bit K_DR key value is successfully retained in the CRYPTOCELL AO power domain

6.6.8.1.1.4 HOST_IOT_KDR1

Address offset: 0x1A54

This register holds bits 63:32 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	HOST_IOT_KDR1			K_DR bits 63:32

6.6.8.1.1.5 HOST_IOT_KDR2

Address offset: 0x1A58

This register holds bits 95:64 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	HOST_IOT_KDR2			K_DR bits 95:64

6.6.8.1.1.6 HOST_IOT_KDR3

Address offset: 0x1A5C

This register holds bits 127:96 of K_DR. The value of this register is saved in the CRYPTOCELL AO power domain.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	HOST_IOT_KDR3			K_DR bits 127:96

6.6.8.1.1.7 HOST_IOT_LCS

Address offset: 0x1A60

Controls lifecycle state (LCS) for CRYPTOCELL subsystem

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LCS			Lifecycle state value. This field is write-once per reset.
			Debug	0	CC310 operates in debug mode
			Secure	2	CC310 operates in secure mode
B	RW	LCS_IS_VALID			This field is read-only and indicates if CRYPTOCELL LCS has been successfully configured since last reset
			Invalid	0	A valid LCS is not yet retained in the CRYPTOCELL AO power domain
			Valid	1	A valid LCS is successfully retained in the CRYPTOCELL AO power domain

6.7 ECB — AES electronic codebook mode encryption

The AES electronic codebook mode encryption (ECB) can be used for a range of cryptographic functions like hash generation, digital signatures, and keystream generation for data encryption/decryption. The ECB encryption block supports 128 bit AES encryption (encryption only, not decryption).

AES ECB operates with EasyDMA access to system Data RAM for in-place operations on cleartext and ciphertext during encryption. ECB uses the same AES core as the CCM and AAR blocks and is an asynchronous operation which may not complete if the AES core is busy.

AES ECB features:

- 128 bit AES encryption
- Supports standard AES ECB block encryption
- Memory pointer support
- DMA data transfer

AES ECB performs a 128 bit AES block encrypt. At the STARTECB task, data and key is loaded into the algorithm by EasyDMA. When output data has been written back to memory, the ENDECB event is triggered.

AES ECB can be stopped by triggering the STOPECB task.

6.7.1 Shared resources

The ECB, CCM, and AAR share the same AES module. The ECB will always have lowest priority and if there is a sharing conflict during encryption, the ECB operation will be aborted and an ERRORECB event will be generated.

6.7.2 EasyDMA

The ECB implements an EasyDMA mechanism for reading and writing to the Data RAM. This DMA cannot access the program memory or any other parts of the memory area except RAM.

If the ECBDATAPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

The EasyDMA will have finished accessing the Data RAM when the ENDECB or ERRORECB is generated.

6.7.3 ECB data structure

Input to the block encrypt and output from the block encrypt are stored in the same data structure. ECBDATAPTR should point to this data structure before STARTECB is initiated.

Property	Address offset	Description
KEY	0	16 byte AES key
CLEARTEXT	16	16 byte AES cleartext input block
CIPHERTEXT	32	16 byte AES ciphertext output block

Table 39: ECB data structure overview

6.7.4 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption	

Table 40: Instances

Register	Offset	Description
TASKS_STARTECB	0x000	Start ECB block encrypt
TASKS_STOPECB	0x004	Abort a possible executing ECB operation
EVENTS_ENDECB	0x100	ECB block encrypt complete
EVENTS_ERRORECB	0x104	ECB block encrypt aborted because of a STOPECB task or due to an error
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ECBDATAPTR	0x504	ECB block encrypt memory pointers

Table 41: Register overview

6.7.4.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENDECB			Write '1' to enable interrupt for ENDECB event See EVENTS_ENDECB
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERRORECB			Write '1' to enable interrupt for ERRORECB event See EVENTS_ERRORECB
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.7.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENDECB			Write '1' to disable interrupt for ENDECB event See EVENTS_ENDECB
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERRORECB			Write '1' to disable interrupt for ERRORECB event See EVENTS_ERRORECB
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.7.4.3 ECBDATAPTR

Address offset: 0x504

ECB block encrypt memory pointers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ECBDATAPTR			Pointer to the ECB data structure (see Table 1 ECB data structure overview)

6.7.5 Electrical specification

6.7.5.1 ECB Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{ECB}	Run time per 16 byte block in all modes			7.2	µs

6.8 EGU — Event generator unit

The Event generator unit (EGU) provides support for inter-layer signaling. This means support for atomic triggering of both CPU execution and hardware tasks from both firmware (by CPU) and hardware (by PPI). This feature can, for instance, be used for triggering CPU execution at a lower priority execution from a higher priority execution, or to handle a peripheral's ISR execution at a lower priority for some of its events. However, triggering any priority from any priority is possible.

Listed here are the main EGU features:

- Enables SW triggering of interrupts
- Separate interrupt vectors for every EGU instance
- Up to 16 separate event flags per interrupt for multiplexing

Each instance of The EGU implements a set of tasks which can individually be triggered to generate the corresponding event, i.e., the corresponding event for TASKS_TRIGGER[n] is EVENTS_TRIGGERED[n].

Refer to Instances on page 135 for a list of the various EGU instances

6.8.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40014000	EGU	EGU0	Event generator unit 0	
0x40015000	EGU	EGU1	Event generator unit 1	
0x40016000	EGU	EGU2	Event generator unit 2	
0x40017000	EGU	EGU3	Event generator unit 3	
0x40018000	EGU	EGU4	Event generator unit 4	
0x40019000	EGU	EGU5	Event generator unit 5	

Table 42: Instances

Register	Offset	Description
TASKS_TRIGGER[0]	0x000	Trigger 0 for triggering the corresponding TRIGGERED[0] event
TASKS_TRIGGER[1]	0x004	Trigger 1 for triggering the corresponding TRIGGERED[1] event
TASKS_TRIGGER[2]	0x008	Trigger 2 for triggering the corresponding TRIGGERED[2] event
TASKS_TRIGGER[3]	0x00C	Trigger 3 for triggering the corresponding TRIGGERED[3] event
TASKS_TRIGGER[4]	0x010	Trigger 4 for triggering the corresponding TRIGGERED[4] event
TASKS_TRIGGER[5]	0x014	Trigger 5 for triggering the corresponding TRIGGERED[5] event
TASKS_TRIGGER[6]	0x018	Trigger 6 for triggering the corresponding TRIGGERED[6] event
TASKS_TRIGGER[7]	0x01C	Trigger 7 for triggering the corresponding TRIGGERED[7] event
TASKS_TRIGGER[8]	0x020	Trigger 8 for triggering the corresponding TRIGGERED[8] event
TASKS_TRIGGER[9]	0x024	Trigger 9 for triggering the corresponding TRIGGERED[9] event
TASKS_TRIGGER[10]	0x028	Trigger 10 for triggering the corresponding TRIGGERED[10] event
TASKS_TRIGGER[11]	0x02C	Trigger 11 for triggering the corresponding TRIGGERED[11] event
TASKS_TRIGGER[12]	0x030	Trigger 12 for triggering the corresponding TRIGGERED[12] event
TASKS_TRIGGER[13]	0x034	Trigger 13 for triggering the corresponding TRIGGERED[13] event
TASKS_TRIGGER[14]	0x038	Trigger 14 for triggering the corresponding TRIGGERED[14] event
TASKS_TRIGGER[15]	0x03C	Trigger 15 for triggering the corresponding TRIGGERED[15] event
EVENTS_TRIGGERED[0]	0x100	Event number 0 generated by triggering the corresponding TRIGGER[0] task
EVENTS_TRIGGERED[1]	0x104	Event number 1 generated by triggering the corresponding TRIGGER[1] task
EVENTS_TRIGGERED[2]	0x108	Event number 2 generated by triggering the corresponding TRIGGER[2] task
EVENTS_TRIGGERED[3]	0x10C	Event number 3 generated by triggering the corresponding TRIGGER[3] task
EVENTS_TRIGGERED[4]	0x110	Event number 4 generated by triggering the corresponding TRIGGER[4] task
EVENTS_TRIGGERED[5]	0x114	Event number 5 generated by triggering the corresponding TRIGGER[5] task
EVENTS_TRIGGERED[6]	0x118	Event number 6 generated by triggering the corresponding TRIGGER[6] task
EVENTS_TRIGGERED[7]	0x11C	Event number 7 generated by triggering the corresponding TRIGGER[7] task
EVENTS_TRIGGERED[8]	0x120	Event number 8 generated by triggering the corresponding TRIGGER[8] task
EVENTS_TRIGGERED[9]	0x124	Event number 9 generated by triggering the corresponding TRIGGER[9] task
EVENTS_TRIGGERED[10]	0x128	Event number 10 generated by triggering the corresponding TRIGGER[10] task
EVENTS_TRIGGERED[11]	0x12C	Event number 11 generated by triggering the corresponding TRIGGER[11] task
EVENTS_TRIGGERED[12]	0x130	Event number 12 generated by triggering the corresponding TRIGGER[12] task
EVENTS_TRIGGERED[13]	0x134	Event number 13 generated by triggering the corresponding TRIGGER[13] task
EVENTS_TRIGGERED[14]	0x138	Event number 14 generated by triggering the corresponding TRIGGER[14] task
EVENTS_TRIGGERED[15]	0x13C	Event number 15 generated by triggering the corresponding TRIGGER[15] task
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt

Table 43: Register overview

6.8.1.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	TRIGGERED0			Enable or disable interrupt for TRIGGERED[0] event See EVENTS_TRIGGERED[0]
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	TRIGGERED1			Enable or disable interrupt for TRIGGERED[1] event See EVENTS_TRIGGERED[1]
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	TRIGGERED2			Enable or disable interrupt for TRIGGERED[2] event See EVENTS_TRIGGERED[2]
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	TRIGGERED3			Enable or disable interrupt for TRIGGERED[3] event See EVENTS_TRIGGERED[3]
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TRIGGERED4			Enable or disable interrupt for TRIGGERED[4] event See EVENTS_TRIGGERED[4]
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TRIGGERED5			Enable or disable interrupt for TRIGGERED[5] event See EVENTS_TRIGGERED[5]
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	TRIGGERED6			Enable or disable interrupt for TRIGGERED[6] event See EVENTS_TRIGGERED[6]
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	TRIGGERED7			Enable or disable interrupt for TRIGGERED[7] event See EVENTS_TRIGGERED[7]
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	TRIGGERED8			Enable or disable interrupt for TRIGGERED[8] event See EVENTS_TRIGGERED[8]
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	TRIGGERED9			Enable or disable interrupt for TRIGGERED[9] event See EVENTS_TRIGGERED[9]
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	TRIGGERED10			Enable or disable interrupt for TRIGGERED[10] event See EVENTS_TRIGGERED[10]
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
L	RW	TRIGGERED11			Enable or disable interrupt for TRIGGERED[11] event See EVENTS_TRIGGERED[11]
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	TRIGGERED12			Enable or disable interrupt for TRIGGERED[12] event See EVENTS_TRIGGERED[12]
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	TRIGGERED13			Enable or disable interrupt for TRIGGERED[13] event See EVENTS_TRIGGERED[13]
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	TRIGGERED14			Enable or disable interrupt for TRIGGERED[14] event See EVENTS_TRIGGERED[14]
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	TRIGGERED15			Enable or disable interrupt for TRIGGERED[15] event See EVENTS_TRIGGERED[15]
			Disabled	0	Disable
			Enabled	1	Enable

6.8.1.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TRIGGERED0			Write '1' to enable interrupt for TRIGGERED[0] event See EVENTS_TRIGGERED[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	TRIGGERED1			Write '1' to enable interrupt for TRIGGERED[1] event See EVENTS_TRIGGERED[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	TRIGGERED2			Write '1' to enable interrupt for TRIGGERED[2] event See EVENTS_TRIGGERED[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
D	RW	TRIGGERED3			Write '1' to enable interrupt for TRIGGERED[3] event See EVENTS_TRIGGERED[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TRIGGERED4			Write '1' to enable interrupt for TRIGGERED[4] event See EVENTS_TRIGGERED[4]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TRIGGERED5			Write '1' to enable interrupt for TRIGGERED[5] event See EVENTS_TRIGGERED[5]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	TRIGGERED6			Write '1' to enable interrupt for TRIGGERED[6] event See EVENTS_TRIGGERED[6]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TRIGGERED7			Write '1' to enable interrupt for TRIGGERED[7] event See EVENTS_TRIGGERED[7]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	TRIGGERED8			Write '1' to enable interrupt for TRIGGERED[8] event See EVENTS_TRIGGERED[8]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TRIGGERED9			Write '1' to enable interrupt for TRIGGERED[9] event See EVENTS_TRIGGERED[9]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	TRIGGERED10			Write '1' to enable interrupt for TRIGGERED[10] event See EVENTS_TRIGGERED[10]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TRIGGERED11			Write '1' to enable interrupt for TRIGGERED[11] event See EVENTS_TRIGGERED[11]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	TRIGGERED12			Write '1' to enable interrupt for TRIGGERED[12] event See EVENTS_TRIGGERED[12]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	TRIGGERED13			Write '1' to enable interrupt for TRIGGERED[13] event See EVENTS_TRIGGERED[13]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	TRIGGERED14			Write '1' to enable interrupt for TRIGGERED[14] event See EVENTS_TRIGGERED[14]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	TRIGGERED15			Write '1' to enable interrupt for TRIGGERED[15] event See EVENTS_TRIGGERED[15]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.8.1.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TRIGGERED0			Write '1' to disable interrupt for TRIGGERED[0] event See EVENTS_TRIGGERED[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	TRIGGERED1			Write '1' to disable interrupt for TRIGGERED[1] event See EVENTS_TRIGGERED[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	TRIGGERED2			Write '1' to disable interrupt for TRIGGERED[2] event See EVENTS_TRIGGERED[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TRIGGERED3			Write '1' to disable interrupt for TRIGGERED[3] event See EVENTS_TRIGGERED[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
E	RW	TRIGGERED4			Write '1' to disable interrupt for TRIGGERED[4] event See EVENTS_TRIGGERED[4]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TRIGGERED5			Write '1' to disable interrupt for TRIGGERED[5] event See EVENTS_TRIGGERED[5]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	TRIGGERED6			Write '1' to disable interrupt for TRIGGERED[6] event See EVENTS_TRIGGERED[6]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TRIGGERED7			Write '1' to disable interrupt for TRIGGERED[7] event See EVENTS_TRIGGERED[7]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	TRIGGERED8			Write '1' to disable interrupt for TRIGGERED[8] event See EVENTS_TRIGGERED[8]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TRIGGERED9			Write '1' to disable interrupt for TRIGGERED[9] event See EVENTS_TRIGGERED[9]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	TRIGGERED10			Write '1' to disable interrupt for TRIGGERED[10] event See EVENTS_TRIGGERED[10]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TRIGGERED11			Write '1' to disable interrupt for TRIGGERED[11] event See EVENTS_TRIGGERED[11]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	TRIGGERED12			Write '1' to disable interrupt for TRIGGERED[12] event See EVENTS_TRIGGERED[12]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
N	RW	TRIGGERED13			Write '1' to disable interrupt for TRIGGERED[13] event See EVENTS_TRIGGERED[13]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	TRIGGERED14			Write '1' to disable interrupt for TRIGGERED[14] event See EVENTS_TRIGGERED[14]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	TRIGGERED15			Write '1' to disable interrupt for TRIGGERED[15] event See EVENTS_TRIGGERED[15]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.8.2 Electrical specification

6.8.2.1 EGU Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{EGU,EVT}$	Latency between setting an EGU event flag and the system setting an interrupt		1		cycles

6.9 GPIO — General purpose input/output

The general purpose input/output pins (GPIOs) are grouped as one or more ports with each port having up to 32 GPIOs.

The number of ports and GPIOs per port might vary with product variant and package. Refer to Registers on page 144 and Pin assignments on page 524 for more information about the number of GPIOs that are supported.

GPIO has the following user-configurable features:

- Up to 32 GPIO pins per GPIO port
- Configurable output drive strength
- Internal pull-up and pull-down resistors
- Wake-up from high or low level triggers on all pins
- Trigger interrupt on state changes on any pin
- All pins can be used by the PPI task/event system
- One or more GPIO outputs can be controlled through PPI and GPIOTE channels
- All pins can be individually mapped to interface blocks for layout flexibility
- GPIO state changes captured on SENSE signal can be stored by LATCH register

The GPIO port peripheral implements up to 32 pins, `PIN0` through `PIN31`. Each of these pins can be individually configured in the PIN_CNF[n] registers (n=0..31).

The following parameters can be configured through these registers:

- Direction
- Drive strength
- Enabling of pull-up and pull-down resistors
- Pin sensing
- Input buffer disconnect
- Analog input (for selected pins)

The PIN_CNF registers are retained registers. See POWER — Power supply on page 60 chapter for more information about retained registers.

6.9.1 Pin configuration

Pins can be individually configured, through the SENSE field in the PIN_CNF[n] register, to detect either a high level or a low level on their input.

When the correct level is detected on any such configured pin, the sense mechanism will set the DETECT signal high. Each pin has a separate DETECT signal. Default behavior, defined by the DETECTMODE register, is that the DETECT signals from all pins in the GPIO port are combined into one common DETECT signal that is routed throughout the system, which then can be utilized by other peripherals. This mechanism is functional in both System ON mode and System OFF mode. See GPIO port and the GPIO pin details on page 142.

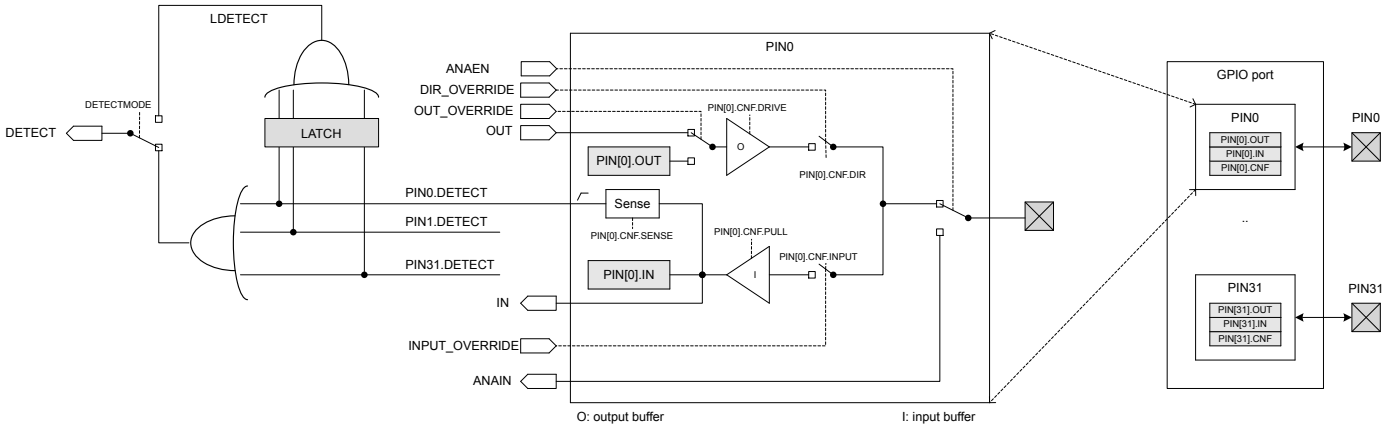

Figure 43: GPIO port and the GPIO pin details

GPIO port and the GPIO pin details on page 142 illustrates the GPIO port containing 32 individual pins, where `PIN0` is illustrated in more detail as a reference. All signals on the left side in the illustration are used by other peripherals in the system and therefore not directly available to the CPU.

Make sure that a pin is in a level that cannot trigger the sense mechanism before enabling it. The DETECT signal will go high immediately if the SENSE condition configured in the PIN_CNF registers is met when the sense mechanism is enabled. This will trigger a PORT event if the DETECT signal was low before enabling the sense mechanism. See GPIOTE — GPIO tasks and events on page 149.

See the following peripherals for more information about how the DETECT signal is used:

- POWER: uses the DETECT signal to exit from System OFF mode.
- GPIOTE: uses the DETECT signal to generate the PORT event.

When a pin's PINx.DETECT signal goes high, a flag will be set in the LATCH register. For example, when the PIN0.DETECT signal goes high, bit 0 in the LATCH register will be set to '1'. If the CPU performs a clear operation on a bit in the LATCH register when the associated PINx.DETECT signal is high, the bit in the LATCH register will not be cleared. The LATCH register will only be cleared if the CPU explicitly clears it by writing a '1' to the bit that shall be cleared, i.e. the LATCH register will not be affected by a PINx.DETECT signal being set low.

The LDETECT signal will be set high when one or more bits in the LATCH register are '1'. The LDETECT signal will be set low when all bits in the LATCH register are successfully cleared to '0'.

If one or more bits in the LATCH register are '1' after the CPU has performed a clear operation on the LATCH registers, a rising edge will be generated on the LDETECT signal. This is illustrated in DETECT signal behavior on page 143.

Important: The CPU can read the LATCH register at any time to check if a SENSE condition has been met on one or more of the the GPIO pins, even if that condition is no longer met at the time the CPU queries the LATCH register. This mechanism will work even if the LDETECT signal is not used as the DETECT signal.

The LDETECT signal is by default not connected to the GPIO port's DETECT signal, but via the DETECTMODE register it is possible to change from default behavior to DETECT signal being derived directly from the LDETECT signal instead. See GPIO port and the GPIO pin details on page 142. DETECT signal behavior on page 143 illustrates the DETECT signal behavior for these two alternatives.

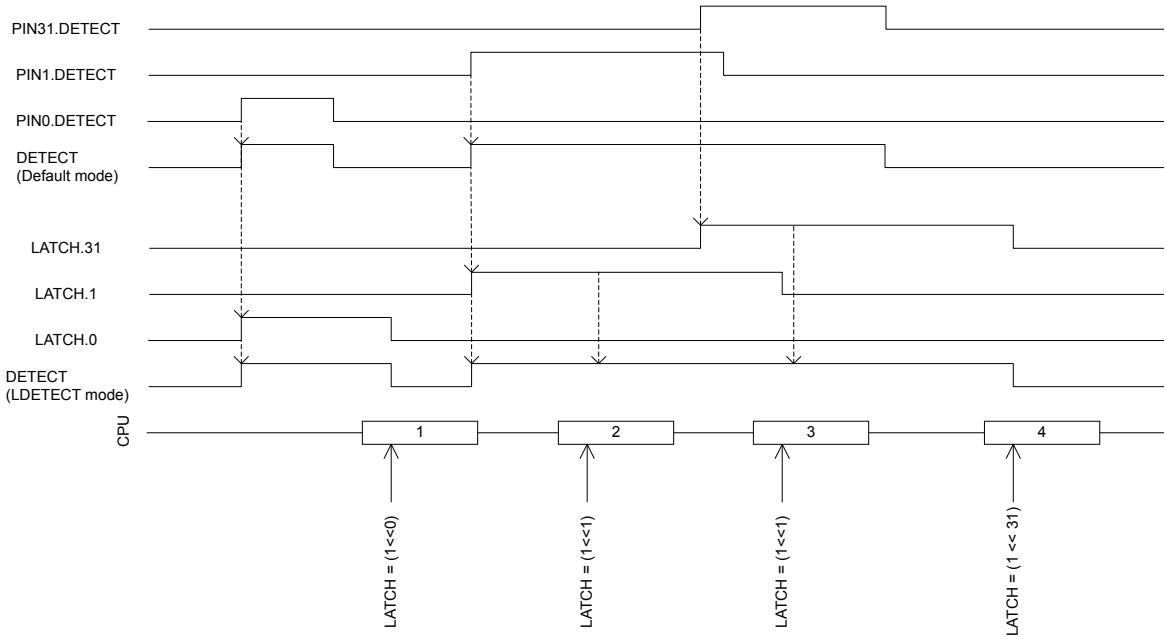

Figure 44: DETECT signal behavior

The input buffer of a GPIO pin can be disconnected from the pin to enable power savings when the pin is not used as an input, see GPIO port and the GPIO pin details on page 142. Inputs must be connected to get a valid input value in the IN register, and for the sense mechanism to get access to the pin.

Other peripherals in the system can connect to GPIO pins and override their output value and configuration, or read their analog or digital input value. See GPIO port and the GPIO pin details on page 142.

Selected pins also support analog input signals, see ANAIN in GPIO port and the GPIO pin details on page 142. The assignment of the analog pins can be found in Pin assignments on page 524.

Important: When a pin is configured as digital input, care has been taken to minimize increased current consumption when the input voltage is between V_{IL} and V_{IH}. However, it is a good practice to ensure that the external circuitry does not drive that pin to levels between V_{IL} and V_{IH} for a long period of time.

6.9.2 Registers

Base address	Peripheral	Instance	Description	Configuration	
0x50000000	GPIO	GPIO	General purpose input and output		Deprecated
0x50000000	GPIO	P0	General purpose input and output, port 0	P0.00 to P0.31 implemented	
0x50000300	GPIO	P1	General purpose input and output, port 1	P1.00 to P1.15 implemented	

Table 44: Instances

Register	Offset	Description
OUT	0x504	Write GPIO port
OUTSET	0x508	Set individual bits in GPIO port
OUTCLR	0x50C	Clear individual bits in GPIO port
IN	0x510	Read GPIO port
DIR	0x514	Direction of GPIO pins
DIRSET	0x518	DIR set register
DIRCLR	0x51C	DIR clear register
LATCH	0x520	Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers
DETECTMODE	0x524	Select between default DETECT signal behaviour and LDETECT mode
PIN_CNF[0]	0x700	Configuration of GPIO pins
PIN_CNF[1]	0x704	Configuration of GPIO pins
PIN_CNF[2]	0x708	Configuration of GPIO pins
PIN_CNF[3]	0x70C	Configuration of GPIO pins
PIN_CNF[4]	0x710	Configuration of GPIO pins
PIN_CNF[5]	0x714	Configuration of GPIO pins
PIN_CNF[6]	0x718	Configuration of GPIO pins
PIN_CNF[7]	0x71C	Configuration of GPIO pins
PIN_CNF[8]	0x720	Configuration of GPIO pins
PIN_CNF[9]	0x724	Configuration of GPIO pins
PIN_CNF[10]	0x728	Configuration of GPIO pins
PIN_CNF[11]	0x72C	Configuration of GPIO pins
PIN_CNF[12]	0x730	Configuration of GPIO pins
PIN_CNF[13]	0x734	Configuration of GPIO pins
PIN_CNF[14]	0x738	Configuration of GPIO pins
PIN_CNF[15]	0x73C	Configuration of GPIO pins
PIN_CNF[16]	0x740	Configuration of GPIO pins
PIN_CNF[17]	0x744	Configuration of GPIO pins
PIN_CNF[18]	0x748	Configuration of GPIO pins
PIN_CNF[19]	0x74C	Configuration of GPIO pins
PIN_CNF[20]	0x750	Configuration of GPIO pins
PIN_CNF[21]	0x754	Configuration of GPIO pins
PIN_CNF[22]	0x758	Configuration of GPIO pins
PIN_CNF[23]	0x75C	Configuration of GPIO pins
PIN_CNF[24]	0x760	Configuration of GPIO pins
PIN_CNF[25]	0x764	Configuration of GPIO pins
PIN_CNF[26]	0x768	Configuration of GPIO pins
PIN_CNF[27]	0x76C	Configuration of GPIO pins
PIN_CNF[28]	0x770	Configuration of GPIO pins
PIN_CNF[29]	0x774	Configuration of GPIO pins
PIN_CNF[30]	0x778	Configuration of GPIO pins

Register	Offset	Description
PIN_CNF[31]	0x77C	Configuration of GPIO pins

Table 45: Register overview

6.9.2.1 OUT

Address offset: 0x504

Write GPIO port

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value																															
A-f	RW	PIN[i] (i=0..31)																																	
			Low	0																															
			High	1																															

6.9.2.2 OUTSET

Address offset: 0x508

Set individual bits in GPIO port

Read: reads value of OUT register.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value																															
A-f	RW	PIN[i] (i=0..31)																																	
			Low	0																															
			High	1																															
			Set	1																															

6.9.2.3 OUTCLR

Address offset: 0x50C

Clear individual bits in GPIO port

Read: reads value of OUT register.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value																															
A-f	RW	PIN[i] (i=0..31)																																	
			Low	0																															
			High	1																															
			Clear	1																															

6.9.2.4 IN

Address offset: 0x510

Read GPIO port

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	R	PIN[i] (i=0..31)			Pin i
			Low	0	Pin input is low
			High	1	Pin input is high

6.9.2.5 DIR

Address offset: 0x514

Direction of GPIO pins

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Pin i
			Input	0	Pin set as input
			Output	1	Pin set as output

6.9.2.6 DIRSET

Address offset: 0x518

DIR set register

Read: reads value of DIR register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Set as output pin i
			Input	0	Read: pin set as input
			Output	1	Read: pin set as output
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect

6.9.2.7 DIRCLR

Address offset: 0x51C

DIR clear register

Read: reads value of DIR register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Set as input pin i
			Input	0	Read: pin set as input
			Output	1	Read: pin set as output
			Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect

6.9.2.8 LATCH

Address offset: 0x520

Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Status on whether PINi has met criteria set in PIN_CNFi.SENSE register. Write '1' to clear.
			NotLatched	0	Criteria has not been met
			Latched	1	Criteria has been met

6.9.2.9 DETECTMODE

Address offset: 0x524

Select between default DETECT signal behaviour and LDETECT mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DETECTMODE			Select between default DETECT signal behaviour and LDETECT mode
			Default	0	DETECT directly connected to PIN DETECT signals
			LDETECT	1	Use the latched LDETECT behaviour

6.9.2.10 PIN_CNF[n] (n=0..31)

Address offset: 0x700 + (n × 0x4)

Configuration of GPIO pins

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E D D D C C B A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
			Input	0	Configure pin as an input pin
			Output	1	Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
			Connect	0	Connect input buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E D D D C C B A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
			Disconnect	1	Disconnect input buffer
C	RW	PULL			Pull configuration
			Disabled	0	No pull
			Pulldown	1	Pull down on pin
			Pullup	3	Pull up on pin
D	RW	DRIVE			Drive configuration
			S0S1	0	Standard '0', standard '1'
			H0S1	1	High drive '0', standard '1'
			S0H1	2	Standard '0', high drive '1'
			H0H1	3	High drive '0', high 'drive '1''
			D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)
			D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)
			S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)
			H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
			Disabled	0	Disabled
			High	2	Sense for high level
			Low	3	Sense for low level

6.9.3 Electrical specification

6.9.3.1 GPIO Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage	0.7 x VDD		VDD	V
V_{IL}	Input low voltage	VSS		0.3 x VDD	V
$V_{OH,SD}$	Output high voltage, standard drive, 0.5 mA, VDD ≥1.7	VDD-0.4		VDD	V
$V_{OH,HDH}$	Output high voltage, high drive, 5 mA, VDD >= 2.7 V	VDD-0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 3 mA, VDD >= 1.7 V	VDD-0.4		VDD	V
$V_{OL,SD}$	Output low voltage, standard drive, 0.5 mA, VDD ≥1.7	VSS		VSS+0.4	V
$V_{OL,HDH}$	Output low voltage, high drive, 5 mA, VDD >= 2.7 V	VSS		VSS+0.4	V
$V_{OL,HDL}$	Output low voltage, high drive, 3 mA, VDD >= 1.7 V	VSS		VSS+0.4	V
$I_{OL,SD}$	Current at VSS+0.4 V, output set low, standard drive, VDD ≥1.7	1	2	4	mA
$I_{OL,HDH}$	Current at VSS+0.4 V, output set low, high drive, VDD >= 2.7 V	6	10	15	mA
$I_{OL,HDL}$	Current at VSS+0.4 V, output set low, high drive, VDD >= 1.7 V	3			mA
$I_{OH,SD}$	Current at VDD-0.4 V, output set high, standard drive, VDD ≥1.7	1	2	4	mA
$I_{OH,HDH}$	Current at VDD-0.4 V, output set high, high drive, VDD >= 2.7 V	6	9	14	mA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{OH,HDL}$	Current at VDD-0.4 V, output set high, high drive, VDD >= 1.7 V	3			mA
$t_{RF,15pF}$	Rise/fall time, standard drive mode, 10-90%, 15 pF load[1]		9		ns
$t_{RF,25pF}$	Rise/fall time, standard drive mode, 10-90%, 25 pF load[1]		13		ns
$t_{RF,50pF}$	Rise/fall time, standard drive mode, 10-90%, 50 pF load[1]		25		ns
$t_{HRF,15pF}$	Rise/Fall time, high drive mode, 10-90%, 15 pF load[1]		4		ns
$t_{HRF,25pF}$	Rise/Fall time, high drive mode, 10-90%, 25 pF load[1]		5		ns
$t_{HRF,50pF}$	Rise/Fall time, high drive mode, 10-90%, 50 pF load[1]		8		ns
R_{PU}	Pull-up resistance	11	13	16	kΩ
R_{PD}	Pull-down resistance	11	13	16	kΩ
C_{PAD}	Pad capacitance		3		pF
C_{PAD_NFC}	Pad capacitance on NFC pads		4		pF
I_{NFC_LEAK}	Leakage current between NFC pads when driven to different states		1	10	µA

6.10 GPIOTE — GPIO tasks and events

The GPIO tasks and events (GPIOTE) module provides functionality for accessing GPIO pins using tasks and events. Each GPIOTE channel can be assigned to one pin.

A GPIOTE block enables GPIOs to generate events on pin state change which can be used to carry out tasks through the PPI system. A GPIO can also be driven to change state on system events using the PPI system. Low power detection of pin state changes is possible when in System ON or System OFF.

Instance	Number of GPIOTE channels
GPIOTE	8

Table 46: GPIOTE properties

Up to three tasks can be used in each GPIOTE channel for performing write operations to a pin. Two tasks are fixed (SET and CLR), and one (OUT) is configurable to perform following operations:

- Set
- Clear
- Toggle

An event can be generated in each GPIOTE channel from one of the following input conditions:

- Rising edge
- Falling edge
- Any change

6.10.1 Pin events and tasks

The GPIOTE module has a number of tasks and events that can be configured to operate on individual GPIO pins.

The tasks (SET[n], CLR[n] and OUT[n]) can be used for writing to individual pins, and the events (IN[n]) can be generated from changes occurring at the inputs of individual pins.

The SET task will set the pin selected in CONFIG[n].PSEL to high.

The CLR task will set the pin low.

[1] Rise and fall times based on simulations

The effect of the OUT task on the pin is configurable in CONFIG[n].POLARITY , and can either set the pin high, set it low, or toggle it.

The tasks and events are configured using the CONFIG[n] registers. Every set of SET, CLR and OUT[n] tasks and IN[n] events has one CONFIG[n] register associated with it.

As long as a SET[n], CLR[n] and OUT[n] task or an IN[n] event is configured to control a pin **n**, the pin's output value will only be updated by the GPIOTE module. The pin's output value as specified in the GPIO will therefore be ignored as long as the pin is controlled by GPIOTE. Attempting to write a pin as a normal GPIO pin will have no effect. When the GPIOTE is disconnected from a pin, see MODE field in CONFIG[n] register, the associated pin will get the output and configuration values specified in the GPIO module.

When conflicting tasks are triggered simultaneously (i.e. during the same clock cycle) in one channel, the precedence of the tasks will be as described in Task priorities on page 150.

Priority	Task
1	OUT
2	CLR
3	SET

Table 47: Task priorities

When setting the CONFIG[n] registers, MODE=Disabled does not have the same effect as MODE=Task and POLARITY=None. In the latter case, a CLR or SET task occurring at the exact same time as OUT will end up with no change on the pin, according to the priorities described in the table above.

When a GPIOTE channel is configured to operate on a pin as a task, the initial value of that pin is configured in the OUTINIT field of CONFIG[n].

6.10.2 Port event

PORT is an event that can be generated from multiple input pins using the GPIO DETECT signal.

The event will be generated on the rising edge of the DETECT signal. See GPIO — General purpose input/output on page 141 for more information about the DETECT signal.

Putting the system into System ON IDLE while DETECT is high will not cause DETECT to wake the system up again. Make sure to clear all DETECT sources before entering sleep. If the LATCH register is used as a source, if any bit in LATCH is still high after clearing all or part of the register (for instance due to one of the PINx.DETECT signal still high), a new rising edge will be generated on DETECT, see Pin configuration on page 142.

Trying to put the system to System OFF while DETECT is high will cause a wakeup from System OFF reset.

This feature is always enabled although the peripheral itself appears to be IDLE, that is, no clocks or other power intensive infrastructure have to be requested to keep this feature enabled. This feature can therefore be used to wake up the CPU from a WFI or WFE type sleep in System ON with all peripherals and the CPU idle, that is, lowest power consumption in System ON mode.

In order to prevent spurious interrupts from the PORT event while configuring the sources, the user shall first disable interrupts on the PORT event (through INTENCLR.PORT), then configure the sources (PIN_CNF[n].SENSE), clear any potential event that could have occurred during configuration (write '0' to EVENTS_PORT), and finally enable interrupts (through INTENSET.PORT).

6.10.3 Tasks and events pin configuration

Each GPIOTE channel is associated with one physical GPIO pin through the CONFIG.PSEL field.

When Event mode is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will be configured as an input, overriding the DIR setting in GPIO. Similarly, when Task mode is selected in CONFIG.MODE,

the pin specified by CONFIG.PSEL will be configured as an output overriding the DIR setting and OUT value in GPIO. When Disabled is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will use its configuration from the PIN[n].CNF registers in GPIO.

Only one GPIOTE channel can be assigned to one physical pin. Failing to do so may result in unpredictable behavior.

6.10.4 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40006000	GPIOTE	GPIOTE	GPIO tasks and events	

Table 48: Instances

Register	Offset	Description
TASKS_OUT[0]	0x000	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is configured in CONFIG[0].POLARITY.
TASKS_OUT[1]	0x004	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is configured in CONFIG[1].POLARITY.
TASKS_OUT[2]	0x008	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is configured in CONFIG[2].POLARITY.
TASKS_OUT[3]	0x00C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is configured in CONFIG[3].POLARITY.
TASKS_OUT[4]	0x010	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is configured in CONFIG[4].POLARITY.
TASKS_OUT[5]	0x014	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is configured in CONFIG[5].POLARITY.
TASKS_OUT[6]	0x018	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is configured in CONFIG[6].POLARITY.
TASKS_OUT[7]	0x01C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is configured in CONFIG[7].POLARITY.
TASKS_SET[0]	0x030	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it high.
TASKS_SET[1]	0x034	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it high.
TASKS_SET[2]	0x038	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it high.
TASKS_SET[3]	0x03C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is to set it high.
TASKS_SET[4]	0x040	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is to set it high.
TASKS_SET[5]	0x044	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is to set it high.
TASKS_SET[6]	0x048	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is to set it high.
TASKS_SET[7]	0x04C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is to set it high.
TASKS_CLR[0]	0x060	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it low.
TASKS_CLR[1]	0x064	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it low.
TASKS_CLR[2]	0x068	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it low.
TASKS_CLR[3]	0x06C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is to set it low.
TASKS_CLR[4]	0x070	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is to set it low.
TASKS_CLR[5]	0x074	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is to set it low.
TASKS_CLR[6]	0x078	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is to set it low.
TASKS_CLR[7]	0x07C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is to set it low.
EVENTS_IN[0]	0x100	Event generated from pin specified in CONFIG[0].PSEL
EVENTS_IN[1]	0x104	Event generated from pin specified in CONFIG[1].PSEL
EVENTS_IN[2]	0x108	Event generated from pin specified in CONFIG[2].PSEL
EVENTS_IN[3]	0x10C	Event generated from pin specified in CONFIG[3].PSEL
EVENTS_IN[4]	0x110	Event generated from pin specified in CONFIG[4].PSEL
EVENTS_IN[5]	0x114	Event generated from pin specified in CONFIG[5].PSEL
EVENTS_IN[6]	0x118	Event generated from pin specified in CONFIG[6].PSEL

Register	Offset	Description
EVENTS_IN[7]	0x11C	Event generated from pin specified in CONFIG[7].PSEL
EVENTS_PORT	0x17C	Event generated from multiple input GPIO pins with SENSE mechanism enabled
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
CONFIG[0]	0x510	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[1]	0x514	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[2]	0x518	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[3]	0x51C	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[4]	0x520	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[5]	0x524	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[6]	0x528	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
CONFIG[7]	0x52C	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Table 49: Register overview

6.10.4.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	IN0			Write '1' to enable interrupt for IN[0] event See EVENTS_IN[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	IN1			Write '1' to enable interrupt for IN[1] event See EVENTS_IN[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	IN2			Write '1' to enable interrupt for IN[2] event See EVENTS_IN[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	IN3			Write '1' to enable interrupt for IN[3] event See EVENTS_IN[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	IN4			Write '1' to enable interrupt for IN[4] event See EVENTS_IN[4]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	IN5			Write '1' to enable interrupt for IN[5] event See EVENTS_IN[5]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	IN6			Write '1' to enable interrupt for IN[6] event See EVENTS_IN[6]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	IN7			Write '1' to enable interrupt for IN[7] event See EVENTS_IN[7]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PORT			Write '1' to enable interrupt for PORT event See EVENTS_PORT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.10.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	IN0			Write '1' to disable interrupt for IN[0] event See EVENTS_IN[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	IN1			Write '1' to disable interrupt for IN[1] event See EVENTS_IN[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	IN2			Write '1' to disable interrupt for IN[2] event See EVENTS_IN[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	IN3			Write '1' to disable interrupt for IN[3] event See EVENTS_IN[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
E	RW	IN4			Write '1' to disable interrupt for IN[4] event See EVENTS_IN[4]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	IN5			Write '1' to disable interrupt for IN[5] event See EVENTS_IN[5]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	IN6			Write '1' to disable interrupt for IN[6] event See EVENTS_IN[6]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	IN7			Write '1' to disable interrupt for IN[7] event See EVENTS_IN[7]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PORT			Write '1' to disable interrupt for PORT event See EVENTS_PORT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.10.4.3 CONFIG[n] (n=0..7)

Address offset: 0x510 + (n × 0x4)

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D D C B B B B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MODE			Mode
			Disabled	0	Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.
			Event	1	Event mode The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D D C B B B B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Task	3	Task mode The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.
B	RW	PSEL		[0..31]	GPIO number associated with SET[n], CLR[n] and OUT[n] tasks and IN[n] event
C	RW	PORT		[0..1]	Port number
D	RW	POLARITY			When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.
			None	0	Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.
			LoToHi	1	Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.
			HiToLo	2	Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.
			Toggle	3	Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.
E	RW	OUTINIT			When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.
			Low	0	Task mode: Initial value of pin before task triggering is low
			High	1	Task mode: Initial value of pin before task triggering is high

6.10.5 Electrical specification

6.11 I²S — Inter-IC sound interface

The I²S (Inter-IC Sound) module, supports the original two-channel I²S format, and left or right-aligned formats. It implements EasyDMA for sample transfer directly to and from RAM without CPU intervention.

The I²S peripheral has the following main features:

- Master and Slave mode
- Simultaneous bi-directional (TX and RX) audio streaming
- Original I²S and left- or right-aligned format
- 8, 16 and 24-bit sample width
- Low-jitter Master Clock generator
- Various sample rates

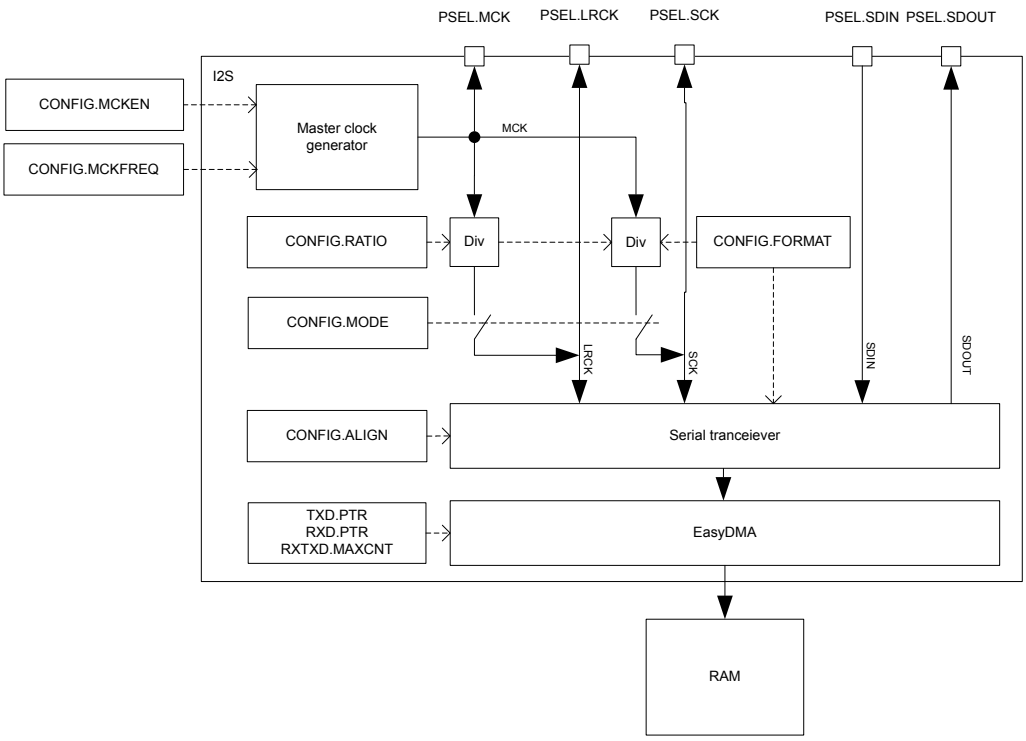

Figure 45: I²S master

6.11.1 Mode

The I²S protocol specification defines two modes of operation, Master and Slave.

The I²S mode decides which of the two sides (Master or Slave) shall provide the clock signals LRCK and SCK, and these signals are always supplied by the Master to the Slave.

6.11.2 Transmitting and receiving

The I²S module supports both transmission (TX) and reception (RX) of serial data. In both cases the serial data is shifted synchronously to the clock signals SCK and LRCK.

TX data is written to the SDOUT pin on the falling edge of SCK, and RX data is read from the SDIN pin on the rising edge of SCK. The most significant bit (MSB) is always transmitted first.

TX and RX are available in both Master and Slave modes and can be enabled/disabled independently in the CONFIG.TXEN on page 167 and CONFIG.RXEN on page 167.

Transmission and/or reception is started by triggering the START task. When started and transmission is enabled (in CONFIG.TXEN on page 167), the TXPTRUPD event will be generated for every RXTXD.MAXCNT on page 170 number of transmitted data words (containing one or more samples). Similarly, when started and reception is enabled (in CONFIG.RXEN on page 167), the RXPTRUPD event will be generated for every RXTXD.MAXCNT on page 170 received data words.

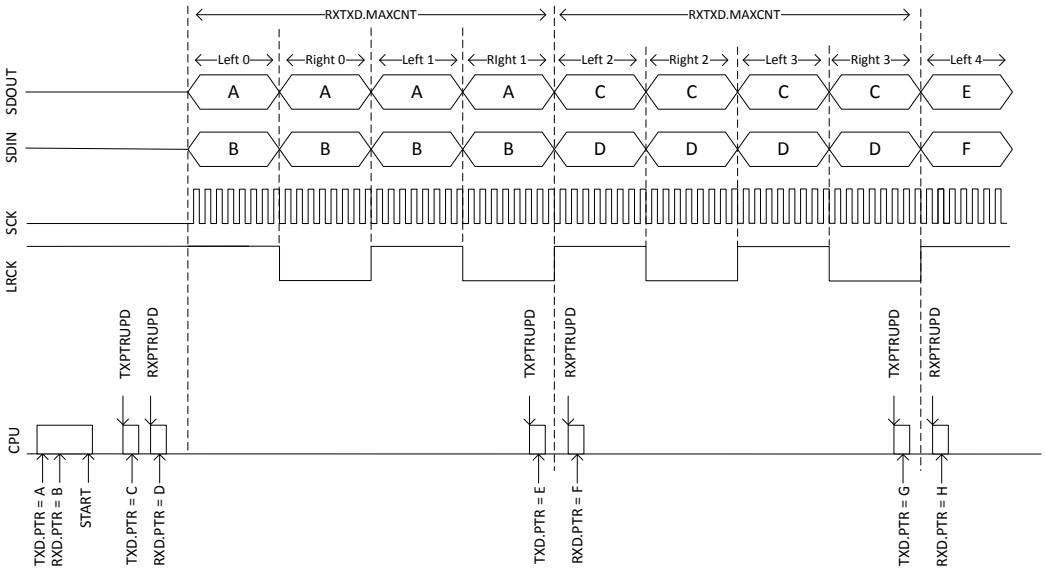

Figure 46: Transmitting and receiving. CONFIG.FORMAT = Aligned, CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo, RXTXD.MAXCNT = 1.

6.11.3 Left right clock (LRCK)

The Left Right Clock (LRCK), often referred to as "word clock", "sample clock" or "word select" in I^2S context, is the clock defining the frames in the serial bit streams sent and received on SDOUT and SDIN, respectively.

In I2S mode, each frame contains one left and right sample pair, with the left sample being transferred during the low half period of LRCK followed by the right sample being transferred during the high period of LRCK.

In Aligned mode, each frame contains one left and right sample pair, with the left sample being transferred during the high half period of LRCK followed by the right sample being transferred during the low period of LRCK.

Consequently, the LRCK frequency is equivalent to the audio sample rate.

When operating in Master mode, the LRCK is generated from the MCK, and the frequency of LRCK is then given as:

```
LRCK = MCK / CONFIG.RATIO
```

LRCK always toggles around the falling edge of the serial clock SCK.

6.11.4 Serial clock (SCK)

The serial clock (SCK), often referred to as the serial bit clock, pulses once for each data bit being transferred on the serial data lines SDIN and SDOUT.

When operating in Master mode the SCK is generated from the MCK, and the frequency of SCK is then given as:

```
SCK = 2 * LRCK * CONFIG.SWIDTH
```

The falling edge of the SCK falls on the toggling edge of LRCK.

When operating in Slave mode SCK is provided by the external I^2S master.

6.11.5 Master clock (MCK)

The master clock (MCK) is the clock from which LRCK and SCK are derived when operating in Master mode.

The MCK is generated by an internal MCK generator. This generator always needs to be enabled when in Master mode, but the generator can also be enabled when in Slave mode. Enabling the generator when in slave mode can be useful in the case where the external Master is not able to generate its own master clock.

The MCK generator is enabled/disabled in the register CONFIG.MCKEN on page 168, and the generator is started or stopped by the START or STOP tasks.

In Master mode the LRCK and the SCK frequencies are closely related, as both are derived from MCK and set indirectly through CONFIG.RATIO on page 168 and CONFIG.SWIDTH on page 169.

When configuring these registers, the user is responsible for fulfilling the following requirements:

1. SCK frequency can never exceed the MCK frequency, which can be formulated as:

```
CONFIG.RATIO >= 2 * CONFIG.SWIDTH
```

2. The MCK/LRCK ratio shall be a multiple of 2 * CONFIG.SWIDTH, which can be formulated as:

```
Integer = (CONFIG.RATIO / (2 * CONFIG.SWIDTH))
```

The MCK signal can be routed to an output pin (specified in PSEL.MCK) to supply external I^{2}S devices that require the MCK to be supplied from the outside.

When operating in Slave mode, the I^{2}S module does not use the MCK and the MCK generator does not need to be enabled.

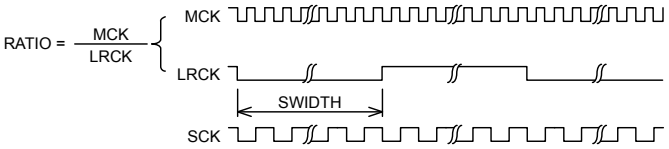

Figure 47: Relation between RATIO, MCK and LRCK.

Desired LRCK [Hz]	CONFIG.SWID	CONFIG.RATIO	CONFIG.MCKF	MCK [Hz]	LRCK [Hz]	LRCK error [%]
16000	16Bit	32X	32MDIV63	507936.5	15873.0	-0.8
16000	16Bit	64X	32MDIV31	1032258.1	16129.0	0.8
16000	16Bit	256X	32MDIV8	4000000.0	15625.0	-2.3
32000	16Bit	32X	32MDIV31	1032258.1	32258.1	0.8
32000	16Bit	64X	32MDIV16	2000000.0	31250.0	-2.3
32000	16Bit	256X	32MDIV4	8000000.0	31250.0	-2.3
44100	16Bit	32X	32MDIV23	1391304.3	43478.3	-1.4
44100	16Bit	64X	32MDIV11	2909090.9	45454.5	3.1
44100	16Bit	256X	32MDIV3	10666666.7	41666.7	-5.5

Table 50: Configuration examples

6.11.6 Width, alignment and format

The CONFIG.SWIDTH register primarily defines the sample width of the data written to memory. In master mode, it then also sets the amount of bits per frame. In Slave mode it controls padding/trimming if required. Left, right, transmitted, and received samples always have the same width. The CONFIG.FORMAT register specifies the position of the data frames with respect to the LRCK edges in both Master and Slave modes.

When using I^2S format, the first bit in a half-frame (containing one left or right sample) gets sampled on the second rising edge of the SCK after a LRCK edge. When using Aligned mode, the first bit in a half-frame gets sampled on the first rising edge of SCK following a LRCK edge.

For data being received on SDIN the sample value can be either right or left-aligned inside a half-frame, as specified in CONFIG.ALIGN on page 169. CONFIG.ALIGN on page 169 affects only the decoding of the incoming samples (SDIN), while the outgoing samples (SDOUT) are always left-aligned (or justified).

When using left-alignment, each half-frame starts with the MSB of the sample value (both for data being sent on SDOUT and received on SDIN).

When using right-alignment, each half-frame of data being received on SDIN ends with the LSB of the sample value, while each half-frame of data being sent on SDOUT starts with the MSB of the sample value (same as for left-alignment).

In Master mode, the size of a half-frame (in number of SCK periods) equals the sample width (in number of bits), and in this case the alignment setting does not care as each half-frame in any case will start with the MSB and end with the LSB of the sample value.

In slave mode, however, the sample width does not need to equal the frame size. This means you might have extra or fewer SCK pulses per half-frame than what the sample width specified in CONFIG.SWIDTH requires.

In the case where we use **left-alignment** and the number of SCK pulses per half-frame is **higher** than the sample width, the following will apply:

- For data received on SDIN, all bits after the LSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0.

In the case where we use **left-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

- Data sent and received on SDOUT and SDIN will be truncated with the LSBs being removed first.

In the case where we use **right-alignment** and the number of SCK pulses per frame is **higher** than the sample width, the following will apply:

- For data received on SDIN, all bits before the MSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0 (same behavior as for left-alignment).

In the case where we use **right-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

- Data received on SDIN will be sign-extended to "sample width" number of bits before being written to memory.
- Data sent on SDOUT will be truncated with the LSBs being removed first (same behavior as for left-alignment).

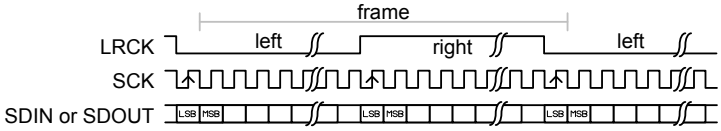

Figure 48: I^2S format. CONFIG.SWIDTH equalling half-frame size.

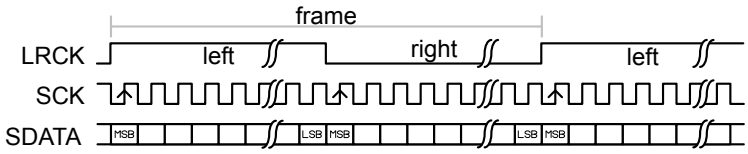

Figure 49: Aligned format. CONFIG.SWIDTH equalling half-frame size.

6.11.7 EasyDMA

The I²S module implements EasyDMA for accessing internal Data RAM without CPU intervention.

The source and destination pointers for the TX and RX data are configured in TXD.PTR on page 170 and RXD.PTR on page 170. The memory pointed to by these pointers will only be read or written when TX or RX are enabled in CONFIG.TXEN on page 167 and CONFIG.RXEN on page 167.

The addresses written to the pointer registers TXD.PTR on page 170 and RXD.PTR on page 170 are double-buffered in hardware, and these double buffers are updated for every RXTXD.MAXCNT on page 170 words (containing one or more samples) read/written from/to memory. The events TXPTRUPD and RXPTRUPD are generated whenever the TXD.PTR and RXD.PTR are transferred to these double buffers.

If TXD.PTR on page 170 is not pointing to the Data RAM region when transmission is enabled, or RXD.PTR on page 170 is not pointing to the Data RAM region when reception is enabled, an EasyDMA transfer may result in a HardFault and/or memory corruption. See Memory on page 19 for more information about the different memory regions.

Due to the nature of I²S, where the number of transmitted samples always equals the number of received samples (at least when both TX and RX are enabled), one common register RXTXD.MAXCNT on page 170 is used for specifying the sizes of these two memory buffers. The size of the buffers is specified in a number of 32-bit words. Such a 32-bit memory word can either contain four 8-bit samples, two 16-bit samples or one right-aligned 24-bit sample sign extended to 32 bit.

In stereo mode (CONFIG.CHANNELS=Stereo), the samples are stored as "left and right sample pairs" in memory. Figure Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo. on page 160, Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo. on page 161 and Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo. on page 161 show how the samples are mapped to memory in this mode. The mapping is valid for both RX and TX.

In mono mode (CONFIG.CHANNELS=Left or Right), RX sample from only one channel in the frame is stored in memory, the other channel sample is ignored. Illustrations Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left. on page 161, Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left. on page 161 and Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left. on page 162 show how RX samples are mapped to memory in this mode.

For TX, the same outgoing sample read from memory is transmitted on both left and right in a frame, resulting in a mono output stream.

	31 24	23 16	15 8	7 0
x.PTR	Right sample 1	Left sample 1	Right sample 0	Left sample 0
x.PTR + 4	Right sample 3	Left sample 3	Right sample 2	Left sample 2
x.PTR + (n*2) - 4	Left sample n-1	Right sample n-1	Right sample n-2	Left sample n-2

Figure 50: Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.

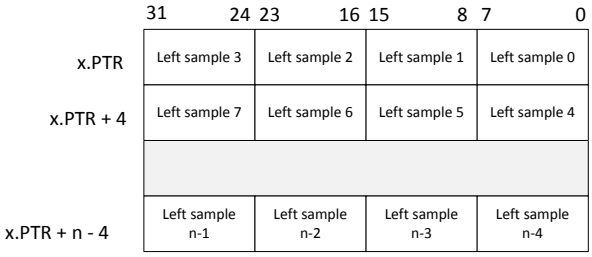

Figure 51: Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.

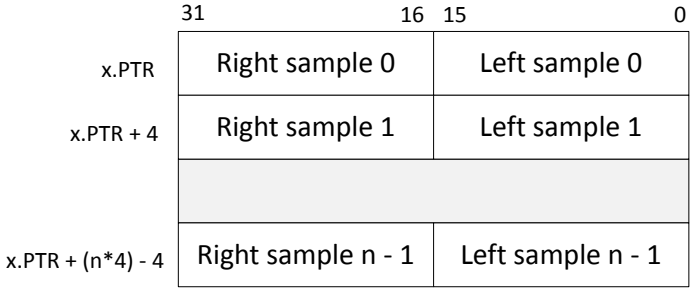

Figure 52: Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.

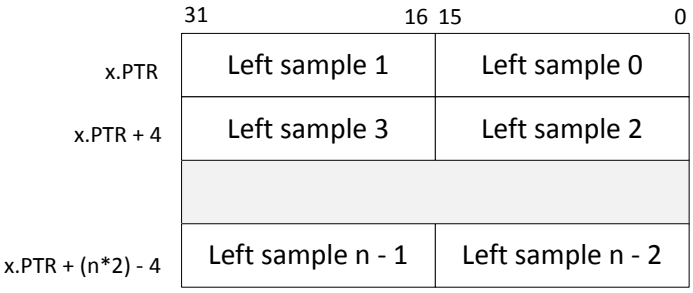

Figure 53: Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.

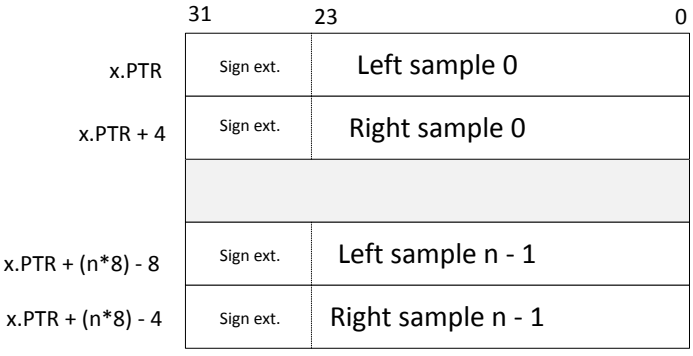

Figure 54: Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.

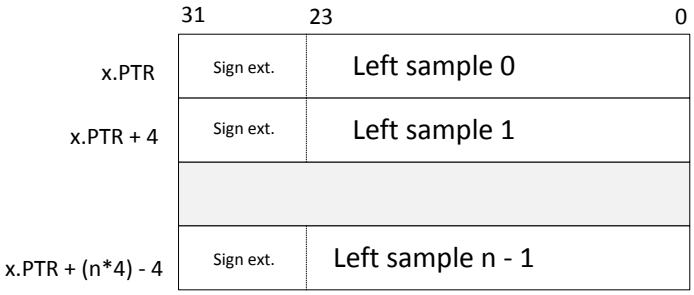

Figure 55: Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.

6.11.8 Module operation

Described here is a typical operating procedure for the I^{2}S module.

1. Configure the I^{2}S module using the CONFIG registers

```
// Enable reception
NRF_I2S->CONFIG.RXEN = (I2S_CONFIG_RXEN_RXEN_Enabled <<
                                        I2S_CONFIG_RXEN_RXEN_Pos);
// Enable transmission
NRF_I2S->CONFIG.TXEN = (I2S_CONFIG_TXEN_TXEN_Enabled <<
                                        I2S_CONFIG_TXEN_TXEN_Pos);
// Enable MCK generator
NRF_I2S->CONFIG.MCKEN = (I2S_CONFIG_MCKEN_MCKEN_Enabled <<
                                        I2S_CONFIG_MCKEN_MCKEN_Pos);
// MCKFREQ = 4 MHz
NRF_I2S->CONFIG.MCKFREQ = I2S_CONFIG_MCKFREQ_MCKFREQ_32MDIV8 <<
                                        I2S_CONFIG_MCKFREQ_MCKFREQ_Pos;
// Ratio = 256
NRF_I2S->CONFIG.RATIO = I2S_CONFIG_RATIO_RATIO_256X <<
                                        I2S_CONFIG_RATIO_RATIO_Pos;
// MCKFREQ = 4 MHz and Ratio = 256 gives sample rate = 15.625 ks/s
// Sample width = 16 bit
NRF_I2S->CONFIG.SWIDTH = I2S_CONFIG_SWIDTH_SWIDTH_16Bit <<
                                        I2S_CONFIG_SWIDTH_SWIDTH_Pos;
// Alignment = Left
NRF_I2S->CONFIG.ALIGN = I2S_CONFIG_ALIGN_ALIGN_Left <<
                                        I2S_CONFIG_ALIGN_ALIGN_Pos;
// Format = I2S
NRF_I2S->CONFIG.FORMAT = I2S_CONFIG_FORMAT_FORMAT_I2S <<
                                        I2S_CONFIG_FORMAT_FORMAT_Pos;
// Use stereo
NRF_I2S->CONFIG.CHANNELS = I2S_CONFIG_CHANNELS_CHANNELS_Stereo <<
                                        I2S_CONFIG_CHANNELS_CHANNELS_Pos;
```

2. Map IO pins using the PINSEL registers

```
// MCK routed to pin 0
NRF_I2S->PSEL.MCK = (0 << I2S_PSEL_MCK_PIN_Pos) |
                    (I2S_PSEL_MCK_CONNECT_Connected <<
                                                I2S_PSEL_MCK_CONNECT_Pos);
// SCK routed to pin 1
NRF_I2S->PSEL.SCK = (1 << I2S_PSEL_SCK_PIN_Pos) |
                    (I2S_PSEL_SCK_CONNECT_Connected <<
                                                I2S_PSEL_SCK_CONNECT_Pos);
// LRCK routed to pin 2
NRF_I2S->PSEL.LRCK = (2 << I2S_PSEL_LRCK_PIN_Pos) |
                     (I2S_PSEL_LRCK_CONNECT_Connected <<
                                                I2S_PSEL_LRCK_CONNECT_Pos);
// SDOUT routed to pin 3
NRF_I2S->PSEL.SDOUT = (3 << I2S_PSEL_SDOUT_PIN_Pos) |
                      (I2S_PSEL_SDOUT_CONNECT_Connected <<
                                                I2S_PSEL_SDOUT_CONNECT_Pos);
// SDIN routed on pin 4
NRF_I2S->PSEL.SDIN = (4 << I2S_PSEL_SDIN_PIN_Pos) |
                     (I2S_PSEL_SDIN_CONNECT_Connected <<
                                                I2S_PSEL_SDIN_CONNECT_Pos);
```

3. Configure TX and RX data pointers using the TXD, RXD and RXTXD registers

```
NRF_I2S->TXD.PTR = my_tx_buf;
NRF_I2S->RXD.PTR = my_rx_buf;
NRF_I2S->TXD.MAXCNT = MY_BUF_SIZE;
```

4. Enable the I^{2}S module using the ENABLE register

```
NRF_I2S->ENABLE = 1;
```

5. Start audio streaming using the START task

```
NRF_I2S->TASKS_START = 1;
```

6. Handle received and transmitted data when receiving the TXPTRUPD and RXPTRUPD events

```
if(NRF_I2S->EVENTS_TXPTRUPD  != 0)
{
    NRF_I2S->TXD.PTR = my_next_tx_buf;
    NRF_I2S->EVENTS_TXPTRUPD = 0;
}

if(NRF_I2S->EVENTS_RXPTRUPD != 0)
{
    NRF_I2S->RXD.PTR = my_next_rx_buf;
    NRF_I2S->EVENTS_RXPTRUPD = 0;
}
```

6.11.9 Pin configuration

The MCK, SCK, LRCK, SDIN and SDOUT signals associated with the I²S module are mapped to physical pins according to the pin numbers specified in the PSEL.x registers.

These pins are acquired whenever the I²S module is enabled through the register ENABLE on page 166.

When a pin is acquired by the I²S module, the direction of the pin (input or output) will be configured automatically, and any pin direction setting done in the GPIO module will be overridden. The directions for the various I²S pins are shown below in GPIO configuration before enabling peripheral (master mode) on page 164 and GPIO configuration before enabling peripheral (slave mode) on page 164.

To secure correct signal levels on the pins when the system is in OFF mode, and when the I²S module is disabled, these pins must be configured in the GPIO peripheral directly.

I²S signal	I²S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Output	0	
SCK	As specified in PSEL.SCK	Output	0	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 51: GPIO configuration before enabling peripheral (master mode)

I²S signal	I²S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 52: GPIO configuration before enabling peripheral (slave mode)

6.11.10 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40025000	I2S	I2S	Inter-IC sound interface	

Table 53: Instances

Register	Offset	Description
TASKS_START	0x000	Starts continuous I2S transfer. Also starts MCK generator when this is enabled.
TASKS_STOP	0x004	Stops I2S transfer. Also stops MCK generator. Triggering this task will cause the {event:STOPPED} event to be generated.
EVENTS_RXPTRUPD	0x104	The RXD.PTR register has been copied to internal double-buffers. When the I2S module is started and RX is enabled, this event will be generated for every RXTXD.MAXCNT words that are received on the SDIN pin.
EVENTS_STOPPED	0x108	I2S transfer stopped.
EVENTS_TXPTRUPD	0x114	The TDX.PTR register has been copied to internal double-buffers. When the I2S module is started and TX is enabled, this event will be generated for every RXTXD.MAXCNT words that are sent on the SDOUT pin.
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt

Register	Offset	Description
ENABLE	0x500	Enable I2S module.
CONFIG.MODE	0x504	I2S mode.
CONFIG.RXEN	0x508	Reception (RX) enable.
CONFIG.TXEN	0x50C	Transmission (TX) enable.
CONFIG.MCKEN	0x510	Master clock generator enable.
CONFIG.MCKFREQ	0x514	Master clock generator frequency.
CONFIG.RATIO	0x518	MCK / LRCK ratio.
CONFIG.SWIDTH	0x51C	Sample width.
CONFIG.ALIGN	0x520	Alignment of sample within a frame.
CONFIG.FORMAT	0x524	Frame format.
CONFIG.CHANNELS	0x528	Enable channels.
RXD.PTR	0x538	Receive buffer RAM start address.
TXD.PTR	0x540	Transmit buffer RAM start address.
RXTXD.MAXCNT	0x550	Size of RXD and TXD buffers.
PSEL.MCK	0x560	Pin select for MCK signal.
PSEL.SCK	0x564	Pin select for SCK signal.
PSEL.LRCK	0x568	Pin select for LRCK signal.
PSEL.SDIN	0x56C	Pin select for SDIN signal.
PSEL.SDOUT	0x570	Pin select for SDOUT signal.

Table 54: Register overview

6.11.10.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					F C B
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Enable or disable interrupt for RXPTRUPD event See EVENTS_RXPTRUPD
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXPTRUPD			Enable or disable interrupt for TXPTRUPD event See EVENTS_TXPTRUPD
			Disabled	0	Disable
			Enabled	1	Enable

6.11.10.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Write '1' to enable interrupt for RXPTRUPD event See EVENTS_RXPTRUPD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXPTRUPD			Write '1' to enable interrupt for TXPTRUPD event See EVENTS_TXPTRUPD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.11.10.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Write '1' to disable interrupt for RXPTRUPD event See EVENTS_RXPTRUPD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXPTRUPD			Write '1' to disable interrupt for TXPTRUPD event See EVENTS_TXPTRUPD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.11.10.4 ENABLE

Address offset: 0x500

Enable I2S module.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable I2S module.
			Disabled	0	Disable
			Enabled	1	Enable

6.11.10.5 CONFIG.MODE

Address offset: 0x504

I2S mode.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MODE			I2S mode.
			Master	0	Master mode. SCK and LRCK generated from internal master clcok (MCK) and output on pins defined by PSEL.xxx.
			Slave	1	Slave mode. SCK and LRCK generated by external master and received on pins defined by PSEL.xxx

6.11.10.6 CONFIG.RXEN

Address offset: 0x508

Reception (RX) enable.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RXEN			Reception (RX) enable.
			Disabled	0	Reception disabled and now data will be written to the RXD.PTR address.
			Enabled	1	Reception enabled.

6.11.10.7 CONFIG.TXEN

Address offset: 0x50C

Transmission (TX) enable.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	TXEN			Transmission (TX) enable.
			Disabled	0	Transmission disabled and now data will be read from the RXD.TXD address.
			Enabled	1	Transmission enabled.

6.11.10.8 CONFIG.MCKEN

Address offset: 0x510

Master clock generator enable.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000001					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	MCKEN			Master clock generator enable.
			Disabled	0	Master clock generator disabled and PSEL.MCK not connected(available as GPIO).
			Enabled	1	Master clock generator running and MCK output on PSEL.MCK.

6.11.10.9 CONFIG.MCKFREQ

Address offset: 0x514

Master clock generator frequency.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x20000000					**0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MCKFREQ			Master clock generator frequency.
			32MDIV2	0x80000000	32 MHz / 2 = 16.0 MHz
			32MDIV3	0x50000000	32 MHz / 3 = 10.6666667 MHz
			32MDIV4	0x40000000	32 MHz / 4 = 8.0 MHz
			32MDIV5	0x30000000	32 MHz / 5 = 6.4 MHz
			32MDIV6	0x28000000	32 MHz / 6 = 5.3333333 MHz
			32MDIV8	0x20000000	32 MHz / 8 = 4.0 MHz
			32MDIV10	0x18000000	32 MHz / 10 = 3.2 MHz
			32MDIV11	0x16000000	32 MHz / 11 = 2.9090909 MHz
			32MDIV15	0x11000000	32 MHz / 15 = 2.1333333 MHz
			32MDIV16	0x10000000	32 MHz / 16 = 2.0 MHz
			32MDIV21	0x0C000000	32 MHz / 21 = 1.5238095
			32MDIV23	0x0B000000	32 MHz / 23 = 1.3913043 MHz
			32MDIV30	0x08800000	32 MHz / 30 = 1.0666667 MHz
			32MDIV31	0x08400000	32 MHz / 31 = 1.0322581 MHz
			32MDIV32	0x08000000	32 MHz / 32 = 1.0 MHz
			32MDIV42	0x06000000	32 MHz / 42 = 0.7619048 MHz
			32MDIV63	0x04100000	32 MHz / 63 = 0.5079365 MHz
			32MDIV125	0x020C0000	32 MHz / 125 = 0.256 MHz

6.11.10.10 CONFIG.RATIO

Address offset: 0x518

MCK / LRCK ratio.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000006				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RATIO			MCK / LRCK ratio.
			32X	0	LRCK = MCK / 32
			48X	1	LRCK = MCK / 48
			64X	2	LRCK = MCK / 64
			96X	3	LRCK = MCK / 96
			128X	4	LRCK = MCK / 128
			192X	5	LRCK = MCK / 192
			256X	6	LRCK = MCK / 256
			384X	7	LRCK = MCK / 384
			512X	8	LRCK = MCK / 512

6.11.10.11 CONFIG.SWIDTH

Address offset: 0x51C

Sample width.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	SWIDTH			Sample width.
			8Bit	0	8 bit.
			16Bit	1	16 bit.
			24Bit	2	24 bit.

6.11.10.12 CONFIG.ALIGN

Address offset: 0x520

Alignment of sample within a frame.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ALIGN			Alignment of sample within a frame.
			Left	0	Left-aligned.
			Right	1	Right-aligned.

6.11.10.13 CONFIG.FORMAT

Address offset: 0x524

Frame format.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FORMAT			Frame format.
			I2S	0	Original I2S format.
			Aligned	1	Alternate (left- or right-aligned) format.

6.11.10.14 CONFIG.CHANNELS

Address offset: 0x528

Enable channels.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CHANNELS			Enable channels.
			Stereo	0	Stereo.
			Left	1	Left only.
			Right	2	Right only.

6.11.10.15 RXD.PTR

Address offset: 0x538

Receive buffer RAM start address.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Receive buffer Data RAM start address. When receiving, words containing samples will be written to this address. This address is a word aligned Data RAM address.

6.11.10.16 TXD.PTR

Address offset: 0x540

Transmit buffer RAM start address.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Transmit buffer Data RAM start address. When transmitting, words containing samples will be fetched from this address. This address is a word aligned Data RAM address.

6.11.10.17 RXTXD.MAXCNT

Address offset: 0x550

Size of RXD and TXD buffers.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT			Size of RXD and TXD buffers in number of 32 bit words.

6.11.10.18 PSEL.MCK

Address offset: 0x560

Pin select for MCK signal.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.11.10.19 PSEL.SCK

Address offset: 0x564

Pin select for SCK signal.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.11.10.20 PSEL.LRCK

Address offset: 0x568

Pin select for LRCK signal.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.11.10.21 PSEL.SDIN

Address offset: 0x56C

Pin select for SDIN signal.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.11.10.22 PSEL.SDOUT

Address offset: 0x570

Pin select for SDOUT signal.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.11.11 Electrical specification

6.11.11.1 I2S timing specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{S_SDIN}	SDIN setup time before SCK rising	20			ns
t_{H_SDIN}	SDIN hold time after SCK rising	15			ns
t_{S_SDOUT}	SDOUT setup time after SCK falling	40			ns
t_{H_SDOUT}	SDOUT hold time before SCK falling	6			ns
t_{SCK_LRCK}	SCLK falling to LRCK edge	-5	0	5	ns
f_{MCK}	MCK frequency			4000	kHz
f_{LRCK}	LRCK frequency			48	kHz
f_{SCK}	SCK frequency			2000	kHz
DC_{CK}	Clock duty cycle (MCK, LRCK, SCK)	45		55	%

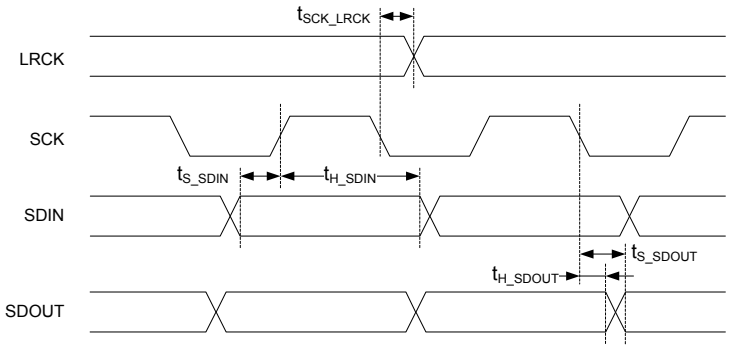

Figure 56: I2S timing diagram

6.12 LPCOMP — Low power comparator

LPCOMP compares an input voltage against a reference voltage.

Listed here are the main features of LPCOMP:

- 0 - VDD input range
- Ultra low power
- Eight input options (`AIN0` to `AIN7`)
- Reference voltage options:
  - Two external analog reference inputs, or
  - 15-level internal reference ladder (VDD/16)
- Optional hysteresis enable on input
- Wakeup source from OFF mode

In System ON, the LPCOMP can generate separate events on rising and falling edges of a signal, or sample the current state of the pin as being above or below the selected reference. The block can be configured to use any of the analog inputs on the device. Additionally, the low power comparator can be used as an analog wakeup source from System OFF or System ON. The comparator threshold can be programmed to a range of fractions of the supply voltage.

Restriction: LPCOMP cannot be used (STARTed) at the same time as COMP. Only one comparator can be used at a time.

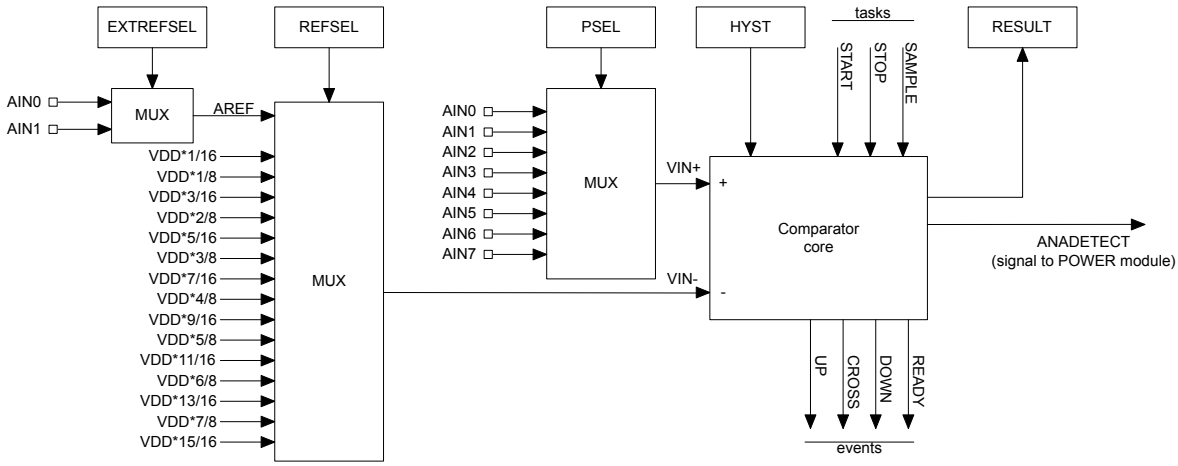

Figure 57: Low power comparator

The wakeup comparator (LPCOMP) compares an input voltage (VIN+), which comes from an analog input pin selected via the PSEL register against a reference voltage (VIN-) selected via the REFSEL on page 178 and EXTREFSEL registers.

The PSEL, REFSEL, and EXTREFSEL registers must be configured before the LPCOMP is enabled through the ENABLE register.

The HYST register allows enabling an optional hysteresis in the comparator core. This hysteresis shall prevent noise on the signal to create unwanted events. See Effect of hysteresis on a noisy input signal on page 174 for illustration of the effect of an active hysteresis on a noisy input signal. It is disabled by default, and shall be configured before enabling LPCOMP as well.

The LPCOMP is started by triggering the START task. After a start-up time of $t_{LPCOMP,STARTUP}$ the LPCOMP will generate a READY event to indicate that the comparator is ready to use and the output of the LPCOMP is correct. The LPCOMP will generate events every time VIN+ crosses VIN-. More specifically, every time VIN+ rises above VIN- (upward crossing) an UP event is generated along with a CROSS event. Every time VIN+ falls below VIN- (downward crossing), a DOWN event is generated along with a CROSS event. When hysteresis is enabled, the upward crossing level becomes (VIN- + VHYST/2), and the downward crossing level becomes (VIN- - VHYST/2).

The LPCOMP is stopped by triggering the STOP task.

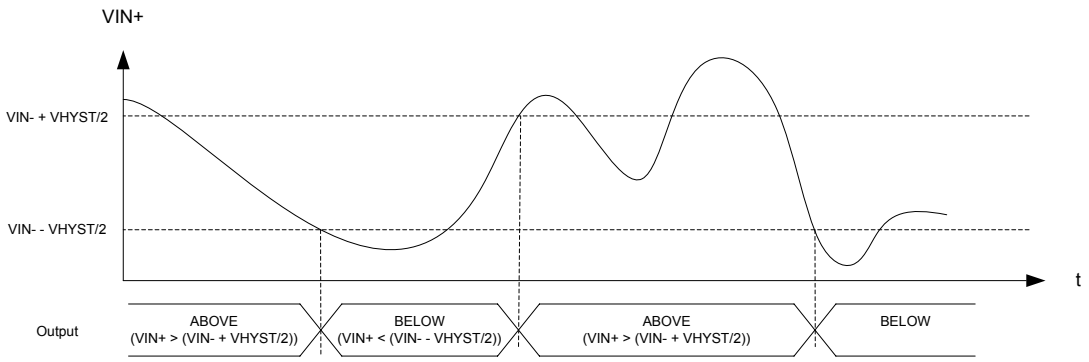

Figure 58: Effect of hysteresis on a noisy input signal

LPCOMP will be operational in both System ON and System OFF mode when it is enabled through the ENABLE register. See POWER — Power supply on page 60 for more information about power modes. Note that it is not allowed to go to System OFF when a READY event is pending to be generated.

All LPCOMP registers, including ENABLE, are classified as retained registers when the LPCOMP is enabled. However, when the device wakes up from System OFF, all LPCOMP registers will be reset.

The LPCOMP can wake up the system from System OFF by asserting the ANADETECT signal. The ANADETECT signal can be derived from any of the event sources that generate the UP, DOWN and CROSS events. In case of wakeup from System OFF, no events will be generated, only the ANADETECT signal. See the ANADETECT register (ANADETECT on page 179) for more information on how to configure the ANADETECT signal.

The immediate value of the LPCOMP can be sampled to RESULT on page 177 by triggering the SAMPLE task.

See RESETREAS on page 72 for more information on how to detect a wakeup from LPCOMP.

6.12.1 Shared resources

The LPCOMP shares resources with other peripherals.

The LPCOMP shares analog resources with SAADC and COMP. While it is possible to use SAADC at the same time as COMP or LPCOMP, COMP and LPCOMP are mutually exclusive: enabling one will

automatically disable the other. In addition, when using SAADC and COMP or LPCOMP simultaneously, it is not possible to select the same analog input pin for both modules.

The LPCOMP peripheral shall not be disabled (by writing to the ENABLE register) before the peripheral has been stopped. Failing to do so may result in unpredictable behaviour.

6.12.2 Pin configuration

You can use the LPCOMP.PSEL register to select one of the analog input pins, `AIN0` through `AIN7`, as the analog input pin for the LPCOMP.

See GPIO — General purpose input/output on page 141 for more information about the pins. Similarly, you can use EXTREFSEL on page 179 to select one of the analog reference input pins, `AIN0` and `AIN1`, as input for AREF in case AREF is selected in EXTREFSEL on page 179. The selected analog pins will be acquired by the LPCOMP when it is enabled through ENABLE on page 178.

6.12.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40013000	LPCOMP	LPCOMP	Low power comparator	

Table 55: Instances

Register	Offset	Description
TASKS_START	0x000	Start comparator
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	LPCOMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RESULT	0x400	Compare result
ENABLE	0x500	Enable LPCOMP
PSEL	0x504	Input pin select
REFSEL	0x508	Reference select
EXTREFSEL	0x50C	External reference select
ANADETECT	0x520	Analog detect configuration
HYST	0x538	Comparator hysteresis enable

Table 56: Register overview

6.12.3.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY_SAMPLE			Shortcut between READY event and SAMPLE task See EVENTS_READY and TASKS_SAMPLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	READY_STOP			Shortcut between READY event and STOP task See EVENTS_READY and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	DOWN_STOP			Shortcut between DOWN event and STOP task See EVENTS_DOWN and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	UP_STOP			Shortcut between UP event and STOP task See EVENTS_UP and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	CROSS_STOP			Shortcut between CROSS event and STOP task See EVENTS_CROSS and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.12.3.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to enable interrupt for DOWN event See EVENTS_DOWN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to enable interrupt for UP event See EVENTS_UP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
D	RW	CROSS			Write '1' to enable interrupt for CROSS event See EVENTS_CROSS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.12.3.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to disable interrupt for DOWN event See EVENTS_DOWN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to disable interrupt for UP event See EVENTS_UP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to disable interrupt for CROSS event See EVENTS_CROSS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.12.3.4 RESULT

Address offset: 0x400

Compare result

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	RESULT			Result of last compare. Decision point SAMPLE task.
			Below	0	Input voltage is below the reference threshold (VIN+ < VIN-).
			Above	1	Input voltage is above the reference threshold (VIN+ > VIN-).

6.12.3.5 ENABLE

Address offset: 0x500

Enable LPCOMP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable LPCOMP
			Disabled	0	Disable
			Enabled	1	Enable

6.12.3.6 PSEL

Address offset: 0x504

Input pin select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PSEL			Analog pin select
			AnalogInput0	0	AIN0 selected as analog input
			AnalogInput1	1	AIN1 selected as analog input
			AnalogInput2	2	AIN2 selected as analog input
			AnalogInput3	3	AIN3 selected as analog input
			AnalogInput4	4	AIN4 selected as analog input
			AnalogInput5	5	AIN5 selected as analog input
			AnalogInput6	6	AIN6 selected as analog input
			AnalogInput7	7	AIN7 selected as analog input

6.12.3.7 REFSEL

Address offset: 0x508

Reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	REFSEL			Reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**
ID	RW	Field	Value ID	Value	Description
			Ref1_8Vdd	0	VDD * 1/8 selected as reference
			Ref2_8Vdd	1	VDD * 2/8 selected as reference
			Ref3_8Vdd	2	VDD * 3/8 selected as reference
			Ref4_8Vdd	3	VDD * 4/8 selected as reference
			Ref5_8Vdd	4	VDD * 5/8 selected as reference
			Ref6_8Vdd	5	VDD * 6/8 selected as reference
			Ref7_8Vdd	6	VDD * 7/8 selected as reference
			ARef	7	External analog reference selected
			Ref1_16Vdd	8	VDD * 1/16 selected as reference
			Ref3_16Vdd	9	VDD * 3/16 selected as reference
			Ref5_16Vdd	10	VDD * 5/16 selected as reference
			Ref7_16Vdd	11	VDD * 7/16 selected as reference
			Ref9_16Vdd	12	VDD * 9/16 selected as reference
			Ref11_16Vdd	13	VDD * 11/16 selected as reference
			Ref13_16Vdd	14	VDD * 13/16 selected as reference
			Ref15_16Vdd	15	VDD * 15/16 selected as reference

6.12.3.8 EXTREFSEL

Address offset: 0x50C

External reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EXTREFSEL			External analog reference select
			AnalogReference0	0	Use AIN0 as external analog reference
			AnalogReference1	1	Use AIN1 as external analog reference

6.12.3.9 ANADETECT

Address offset: 0x520

Analog detect configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ANADETECT			Analog detect configuration
			Cross	0	Generate ANADETECT on crossing, both upward crossing and downward crossing
			Up	1	Generate ANADETECT on upward crossing only
			Down	2	Generate ANADETECT on downward crossing only

6.12.3.10 HYST

Address offset: 0x538

Comparator hysteresis enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	HYST			Comparator hysteresis enable
			Disabled	0	Comparator hysteresis disabled
			Enabled	1	Comparator hysteresis enabled

6.12.4 Electrical specification

6.12.4.1 LPCOMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{LPCANADET}$	Time from VIN crossing (>=50mV above threshold) to ANADETECT signal generated.		5		µs
$V_{INPOFFSET}$	Input offset including reference ladder error	-40		40	mV
V_{HYST}	Optional hysteresis		35		mV
$t_{STARTUP}$	Startup time for LPCOMP		140		µs

6.13 MWU — Memory watch unit

The Memory watch unit (MWU) can be used to generate events when a memory region is accessed by the CPU. The MWU can be configured to trigger events for access to Data RAM and Peripheral memory segments. The MWU allows an application developer to generate memory access events during development for debugging or during production execution for failure detection and recovery.

Listed here are the main features for MWU:

- Six memory regions, four user-configurable and two fixed regions in peripheral address space
- Flexible configuration of regions with START and END addresses
- Generate events on CPU read and/or write to a defined region of Data RAM or peripheral memory address space
- Programmable maskable or non-maskable (NMI) interrupt on events
- Peripheral interfaces can be watched for read and write access using subregions of the two fixed memory regions

Memory region	START address	END address
REGION[0..3]	Configurable	Configurable
PREGION[0]	0x40000000	0x4001FFFF
PREGION[1]	0x40020000	0x4003FFFF

Table 57: Memory regions

Each MWU region is defined by a start address and an end address, configured by the START and END registers respectively. These addresses are byte aligned and inclusive. The END register value has to be greater or equal to the START register value. Each region is associated with a pair of events that indicate that either a write access or a read access from the CPU has been detected inside the region.

For regions containing subregions (see below), a set of status registers PERREGION[0..1].SUBSTATWA and PERREGION[0..1].SUBSTATRA indicate which subregion(s) caused the EVENT_PREGION[0..1].WA and EVENT_PREGION[0..1].RA respectively.

The MWU is only able to detect memory accesses in the Data RAM and Peripheral memory segments from the CPU, see Memory on page 19 for more information about the different memory segments. EasyDMA

accesses are not monitored by the MWU. The MWU requires two HCLK cycles to detect and generate the event.

The peripheral regions, PREGION[0...1], are divided into 32 equally sized subregions, SR[0...31]. All subregions are excluded in the main region by default, and any can be included by specifying them in the SUBS register. When a subregion is excluded from the main region, the memory watch mechanism will not trigger any events when that subregion is accessed.

Subregions in PREGION[0..1] cannot be individually configured for read or write access watch. Watch configuration is only possible for a region as a whole. The PRGNiRA and PRGNiWA (i=0..1) fields in the REGIONEN register control watching read and write access.

REGION[0..3] can be individually enabled for read and/or write access watching through their respective RGNiRA and RGNiWA (i=0..3) fields in the REGIONEN register.

REGIONENSET and REGIONENCLR allow respectively enabling and disabling one or multiple REGIONs or PREGIONs watching in a single write access.

6.13.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40020000	MWU	MWU	Memory watch unit	

Table 58: Instances

Register	Offset	Description
EVENTS_REGION[0].WA	0x100	Write access to region 0 detected
EVENTS_REGION[0].RA	0x104	Read access to region 0 detected
EVENTS_REGION[1].WA	0x108	Write access to region 1 detected
EVENTS_REGION[1].RA	0x10C	Read access to region 1 detected
EVENTS_REGION[2].WA	0x110	Write access to region 2 detected
EVENTS_REGION[2].RA	0x114	Read access to region 2 detected
EVENTS_REGION[3].WA	0x118	Write access to region 3 detected
EVENTS_REGION[3].RA	0x11C	Read access to region 3 detected
EVENTS_PREGION[0].WA	0x160	Write access to peripheral region 0 detected
EVENTS_PREGION[0].RA	0x164	Read access to peripheral region 0 detected
EVENTS_PREGION[1].WA	0x168	Write access to peripheral region 1 detected
EVENTS_PREGION[1].RA	0x16C	Read access to peripheral region 1 detected
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
NMIEN	0x320	Enable or disable non-maskable interrupt
NMIENSET	0x324	Enable non-maskable interrupt
NMIENCLR	0x328	Disable non-maskable interrupt
PERREGION[0].SUBSTATWA	0x400	Source of event/interrupt in region 0, write access detected while corresponding subregion was enabled for watching
PERREGION[0].SUBSTATRA	0x404	Source of event/interrupt in region 0, read access detected while corresponding subregion was enabled for watching
PERREGION[1].SUBSTATWA	0x408	Source of event/interrupt in region 1, write access detected while corresponding subregion was enabled for watching
PERREGION[1].SUBSTATRA	0x40C	Source of event/interrupt in region 1, read access detected while corresponding subregion was enabled for watching
REGIONEN	0x510	Enable/disable regions watch
REGIONENSET	0x514	Enable regions watch
REGIONENCLR	0x518	Disable regions watch

Register	Offset	Description
REGION[0].START	0x600	Start address for region 0
REGION[0].END	0x604	End address of region 0
REGION[1].START	0x610	Start address for region 1
REGION[1].END	0x614	End address of region 1
REGION[2].START	0x620	Start address for region 2
REGION[2].END	0x624	End address of region 2
REGION[3].START	0x630	Start address for region 3
REGION[3].END	0x634	End address of region 3
PREGION[0].START	0x6C0	Reserved for future use
PREGION[0].END	0x6C4	Reserved for future use
PREGION[0].SUBS	0x6C8	Subregions of region 0
PREGION[1].START	0x6D0	Reserved for future use
PREGION[1].END	0x6D4	Reserved for future use
PREGION[1].SUBS	0x6D8	Subregions of region 1

Table 59: Register overview

6.13.1.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID								L	K	J	I																	H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value																															
A	RW	REGION0WA																																	
			Disabled	0																															
			Enabled	1																															
B	RW	REGION0RA																																	
			Disabled	0																															
			Enabled	1																															
C	RW	REGION1WA																																	
			Disabled	0																															
			Enabled	1																															
D	RW	REGION1RA																																	
			Disabled	0																															
			Enabled	1																															
E	RW	REGION2WA																																	
			Disabled	0																															
			Enabled	1																															
F	RW	REGION2RA																																	
			Disabled	0																															
			Enabled	1																															

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
G	RW	REGION3WA			Enable or disable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	REGION3RA			Enable or disable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	PREGION0WA			Enable or disable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	PREGION0RA			Enable or disable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	PREGION1WA			Enable or disable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	PREGION1RA			Enable or disable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Disabled	0	Disable
			Enabled	1	Enable

6.13.1.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REGION0WA			Write '1' to enable interrupt for REGION[0].WA event See EVENTS_REGION[0].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REGION0RA			Write '1' to enable interrupt for REGION[0].RA event See EVENTS_REGION[0].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	REGION1WA			Write '1' to enable interrupt for REGION[1].WA event See EVENTS_REGION[1].WA

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	REGION1RA			Write '1' to enable interrupt for REGION[1].RA event See EVENTS_REGION[1].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	REGION2WA			Write '1' to enable interrupt for REGION[2].WA event See EVENTS_REGION[2].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	REGION2RA			Write '1' to enable interrupt for REGION[2].RA event See EVENTS_REGION[2].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	REGION3WA			Write '1' to enable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	REGION3RA			Write '1' to enable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PREGION0WA			Write '1' to enable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	PREGION0RA			Write '1' to enable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	PREGION1WA			Write '1' to enable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	PREGION1RA			Write '1' to enable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					L K J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled

6.13.1.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					L K J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REGION0WA			Write '1' to disable interrupt for REGION[0].WA event See EVENTS_REGION[0].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REGION0RA			Write '1' to disable interrupt for REGION[0].RA event See EVENTS_REGION[0].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	REGION1WA			Write '1' to disable interrupt for REGION[1].WA event See EVENTS_REGION[1].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	REGION1RA			Write '1' to disable interrupt for REGION[1].RA event See EVENTS_REGION[1].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	REGION2WA			Write '1' to disable interrupt for REGION[2].WA event See EVENTS_REGION[2].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	REGION2RA			Write '1' to disable interrupt for REGION[2].RA event See EVENTS_REGION[2].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	REGION3WA			Write '1' to disable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
H	RW	REGION3RA			Write '1' to disable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PREGION0WA			Write '1' to disable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	PREGION0RA			Write '1' to disable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	PREGION1WA			Write '1' to disable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	PREGION1RA			Write '1' to disable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.13.1.4 NMIEN

Address offset: 0x320

Enable or disable non-maskable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REGION0WA			Enable or disable non-maskable interrupt for REGION[0].WA event See EVENTS_REGION[0].WA
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	REGION0RA			Enable or disable non-maskable interrupt for REGION[0].RA event See EVENTS_REGION[0].RA
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
C	RW	REGION1WA			Enable or disable non-maskable interrupt for REGION[1].WA event See EVENTS_REGION[1].WA
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	REGION1RA			Enable or disable non-maskable interrupt for REGION[1].RA event See EVENTS_REGION[1].RA
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	REGION2WA			Enable or disable non-maskable interrupt for REGION[2].WA event See EVENTS_REGION[2].WA
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	REGION2RA			Enable or disable non-maskable interrupt for REGION[2].RA event See EVENTS_REGION[2].RA
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	REGION3WA			Enable or disable non-maskable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	REGION3RA			Enable or disable non-maskable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	PREGION0WA			Enable or disable non-maskable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	PREGION0RA			Enable or disable non-maskable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	PREGION1WA			Enable or disable non-maskable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
L	RW	PREGION1RA			Enable or disable non-maskable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Disabled	0	Disable
			Enabled	1	Enable

6.13.1.5 NMIENSET

Address offset: 0x324

Enable non-maskable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REGION0WA			Write '1' to enable non-maskable interrupt for REGION[0].WA event See EVENTS_REGION[0].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REGION0RA			Write '1' to enable non-maskable interrupt for REGION[0].RA event See EVENTS_REGION[0].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	REGION1WA			Write '1' to enable non-maskable interrupt for REGION[1].WA event See EVENTS_REGION[1].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	REGION1RA			Write '1' to enable non-maskable interrupt for REGION[1].RA event See EVENTS_REGION[1].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	REGION2WA			Write '1' to enable non-maskable interrupt for REGION[2].WA event See EVENTS_REGION[2].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
F	RW	REGION2RA			Write '1' to enable non-maskable interrupt for REGION[2].RA event See EVENTS_REGION[2].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	REGION3WA			Write '1' to enable non-maskable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	REGION3RA			Write '1' to enable non-maskable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PREGION0WA			Write '1' to enable non-maskable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	PREGION0RA			Write '1' to enable non-maskable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	PREGION1WA			Write '1' to enable non-maskable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	PREGION1RA			Write '1' to enable non-maskable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.13.1.6 NMIENCLR

Address offset: 0x328

Disable non-maskable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID						L	K	J	I																H	G	F	E	D	C	B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	REGION0WA			Write '1' to disable non-maskable interrupt for REGION[0].WA event See EVENTS_REGION[0].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REGION0RA			Write '1' to disable non-maskable interrupt for REGION[0].RA event See EVENTS_REGION[0].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	REGION1WA			Write '1' to disable non-maskable interrupt for REGION[1].WA event See EVENTS_REGION[1].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	REGION1RA			Write '1' to disable non-maskable interrupt for REGION[1].RA event See EVENTS_REGION[1].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	REGION2WA			Write '1' to disable non-maskable interrupt for REGION[2].WA event See EVENTS_REGION[2].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	REGION2RA			Write '1' to disable non-maskable interrupt for REGION[2].RA event See EVENTS_REGION[2].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	REGION3WA			Write '1' to disable non-maskable interrupt for REGION[3].WA event See EVENTS_REGION[3].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	REGION3RA			Write '1' to disable non-maskable interrupt for REGION[3].RA event See EVENTS_REGION[3].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
I	RW	PREGION0WA			Write '1' to disable non-maskable interrupt for PREGION[0].WA event See EVENTS_PREGION[0].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	PREGION0RA			Write '1' to disable non-maskable interrupt for PREGION[0].RA event See EVENTS_PREGION[0].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	PREGION1WA			Write '1' to disable non-maskable interrupt for PREGION[1].WA event See EVENTS_PREGION[1].WA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	PREGION1RA			Write '1' to disable non-maskable interrupt for PREGION[1].RA event See EVENTS_PREGION[1].RA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.13.1.7 PERREGION[n].SUBSTATWA (n=0..1)

Address offset: 0x400 + (n × 0x8)

Source of event/interrupt in region n, write access detected while corresponding subregion was enabled for watching

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	SR[i] (i=0..31)			Subregion i in region n (write '1' to clear)
			NoAccess	0	No write access occurred in this subregion
			Access	1	Write access(es) occurred in this subregion

6.13.1.8 PERREGION[n].SUBSTATRA (n=0..1)

Address offset: 0x404 + (n × 0x8)

Source of event/interrupt in region n, read access detected while corresponding subregion was enabled for watching

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	SR[i] (i=0..31)			Subregion i in region n (write '1' to clear)
			NoAccess	0	No read access occurred in this subregion
			Access	1	Read access(es) occurred in this subregion

6.13.1.9 REGIONEN

Address offset: 0x510

Enable/disable regions watch

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RGN0WA			Enable/disable write access watch in region[0]
			Disable	0	Disable write access watch in this region
			Enable	1	Enable write access watch in this region
B	RW	RGN0RA			Enable/disable read access watch in region[0]
			Disable	0	Disable read access watch in this region
			Enable	1	Enable read access watch in this region
C	RW	RGN1WA			Enable/disable write access watch in region[1]
			Disable	0	Disable write access watch in this region
			Enable	1	Enable write access watch in this region
D	RW	RGN1RA			Enable/disable read access watch in region[1]
			Disable	0	Disable read access watch in this region
			Enable	1	Enable read access watch in this region
E	RW	RGN2WA			Enable/disable write access watch in region[2]
			Disable	0	Disable write access watch in this region
			Enable	1	Enable write access watch in this region
F	RW	RGN2RA			Enable/disable read access watch in region[2]
			Disable	0	Disable read access watch in this region
			Enable	1	Enable read access watch in this region
G	RW	RGN3WA			Enable/disable write access watch in region[3]
			Disable	0	Disable write access watch in this region
			Enable	1	Enable write access watch in this region
H	RW	RGN3RA			Enable/disable read access watch in region[3]
			Disable	0	Disable read access watch in this region
			Enable	1	Enable read access watch in this region
I	RW	PRGN0WA			Enable/disable write access watch in PREGION[0]
			Disable	0	Disable write access watch in this PREGION
			Enable	1	Enable write access watch in this PREGION
J	RW	PRGN0RA			Enable/disable read access watch in PREGION[0]
			Disable	0	Disable read access watch in this PREGION
			Enable	1	Enable read access watch in this PREGION
K	RW	PRGN1WA			Enable/disable write access watch in PREGION[1]
			Disable	0	Disable write access watch in this PREGION
			Enable	1	Enable write access watch in this PREGION
L	RW	PRGN1RA			Enable/disable read access watch in PREGION[1]
			Disable	0	Disable read access watch in this PREGION
			Enable	1	Enable read access watch in this PREGION

6.13.1.10 REGIONENSET

Address offset: 0x514

Enable regions watch

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	RGN0WA			Enable write access watch in region[0]
			Set	1	Enable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
B	RW	RGN0RA			Enable read access watch in region[0]
			Set	1	Enable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
C	RW	RGN1WA			Enable write access watch in region[1]
			Set	1	Enable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
D	RW	RGN1RA			Enable read access watch in region[1]
			Set	1	Enable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
E	RW	RGN2WA			Enable write access watch in region[2]
			Set	1	Enable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
F	RW	RGN2RA			Enable read access watch in region[2]
			Set	1	Enable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
G	RW	RGN3WA			Enable write access watch in region[3]
			Set	1	Enable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
H	RW	RGN3RA			Enable read access watch in region[3]
			Set	1	Enable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
I	RW	PRGN0WA			Enable write access watch in PREGION[0]
			Set	1	Enable write access watch in this PREGION
			Disabled	0	Write access watch in this PREGION is disabled
			Enabled	1	Write access watch in this PREGION is enabled
J	RW	PRGN0RA			Enable read access watch in PREGION[0]
			Set	1	Enable read access watch in this PREGION
			Disabled	0	Read access watch in this PREGION is disabled
			Enabled	1	Read access watch in this PREGION is enabled
K	RW	PRGN1WA			Enable write access watch in PREGION[1]
			Set	1	Enable write access watch in this PREGION
			Disabled	0	Write access watch in this PREGION is disabled
			Enabled	1	Write access watch in this PREGION is enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
L	RW	PRGN1RA			Enable read access watch in PREGION[1]
			Set	1	Enable read access watch in this PREGION
			Disabled	0	Read access watch in this PREGION is disabled
			Enabled	1	Read access watch in this PREGION is enabled

6.13.1.11 REGIONENCLR

Address offset: 0x518

Disable regions watch

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RGN0WA			Disable write access watch in region[0]
			Clear	1	Disable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
B	RW	RGN0RA			Disable read access watch in region[0]
			Clear	1	Disable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
C	RW	RGN1WA			Disable write access watch in region[1]
			Clear	1	Disable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
D	RW	RGN1RA			Disable read access watch in region[1]
			Clear	1	Disable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
E	RW	RGN2WA			Disable write access watch in region[2]
			Clear	1	Disable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
F	RW	RGN2RA			Disable read access watch in region[2]
			Clear	1	Disable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
G	RW	RGN3WA			Disable write access watch in region[3]
			Clear	1	Disable write access watch in this region
			Disabled	0	Write access watch in this region is disabled
			Enabled	1	Write access watch in this region is enabled
H	RW	RGN3RA			Disable read access watch in region[3]
			Clear	1	Disable read access watch in this region
			Disabled	0	Read access watch in this region is disabled
			Enabled	1	Read access watch in this region is enabled
I	RW	PRGN0WA			Disable write access watch in PREGION[0]
			Clear	1	Disable write access watch in this PREGION
			Disabled	0	Write access watch in this PREGION is disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Write access watch in this PREGION is enabled
J	RW	PRGN0RA			Disable read access watch in PREGION[0]
			Clear	1	Disable read access watch in this PREGION
			Disabled	0	Read access watch in this PREGION is disabled
			Enabled	1	Read access watch in this PREGION is enabled
K	RW	PRGN1WA			Disable write access watch in PREGION[1]
			Clear	1	Disable write access watch in this PREGION
			Disabled	0	Write access watch in this PREGION is disabled
			Enabled	1	Write access watch in this PREGION is enabled
L	RW	PRGN1RA			Disable read access watch in PREGION[1]
			Clear	1	Disable read access watch in this PREGION
			Disabled	0	Read access watch in this PREGION is disabled
			Enabled	1	Read access watch in this PREGION is enabled

6.13.1.12 REGION[n].START (n=0..3)

Address offset: 0x600 + (n × 0x10)

Start address for region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	START			Start address for region

6.13.1.13 REGION[n].END (n=0..3)

Address offset: 0x604 + (n × 0x10)

End address of region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END			End address of region.

6.13.1.14 PREGION[n].START (n=0..1)

Address offset: 0x6C0 + (n × 0x10)

Reserved for future use

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	START			Reserved for future use

6.13.1.15 PREGION[n].END (n=0..1)

Address offset: 0x6C4 + (n × 0x10)

Reserved for future use

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	END			Reserved for future use

6.13.1.16 PREGION[n].SUBS (n=0..1)

Address offset: 0x6C8 + (n × 0x10)

Subregions of region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	SR[i] (i=0..31)			Include or exclude subregion i in region
			Exclude	0	Exclude
			Include	1	Include

6.14 NFCT — Near field communication tag

The NFCT peripheral is an implementation of an NFC Forum compliant listening device NFC-A.

With appropriate software, the NFCT peripheral can be used as the listening device NFC-A as specified by the NFC Forum.

Listed here are the main features for the NFCT peripheral:

- NFC-A listen mode operation
  - 13.56 MHz input frequency
  - Bit rate 106 kbps
- Wake-on-field low power field detection (SENSE) mode
- Frame assemble and disassemble for the NFC-A frames specified by the NFC Forum
- Programmable frame timing controller
- Integrated automatic collision resolution, cyclic redundancy check (CRC), and parity functions

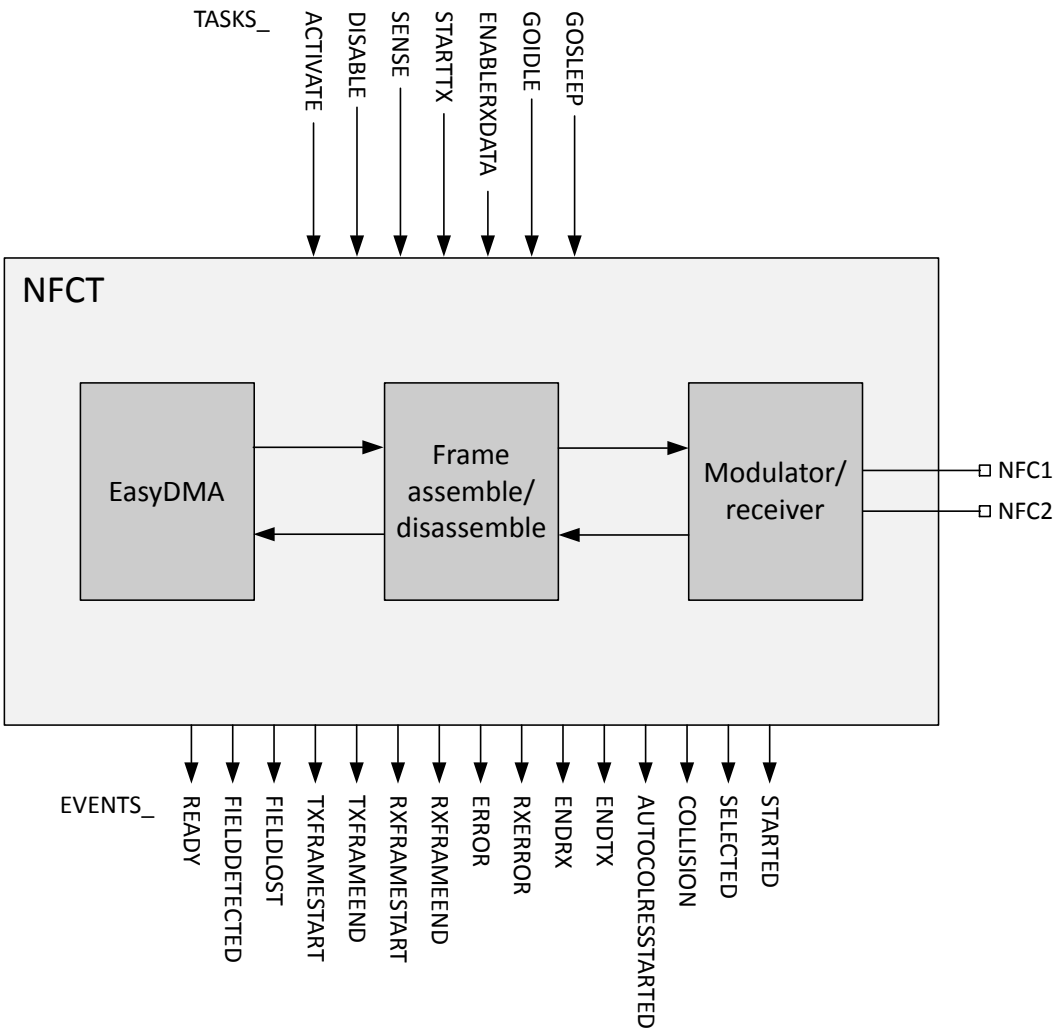

Figure 59: NFCT block diagram

6.14.1 Overview

The NFCT peripheral contains a 13.56 MHz AM receiver and a 13.56 MHz load modulator with 106 kbps data rate as defined by the NFC Forum.

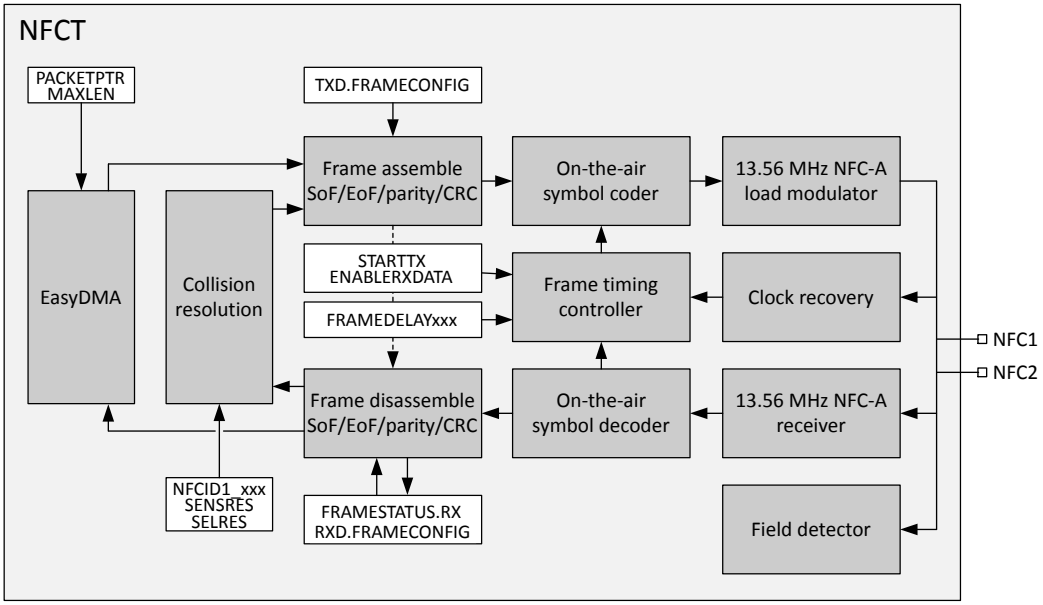

Figure 60: NFCT overview

When transmitting, the frame data will be transferred directly from RAM and transmitted with configurable frame type and delay timing. The system will be notified by an event whenever a complete frame is received or sent. The received frames will be automatically disassembled and the data part of the frame transferred to RAM.

The NFCT peripheral also supports the collision detection and resolution ("anticollision") as defined by the NFC Forum.

Wake-on-field is supported in SENSE mode while the device is either in System OFF or System ON mode. When the antenna enters an NFC field, an event will be triggered notifying the system to activate the NFCT functionality for incoming frames. In System ON, if the energy detected at the antenna increases beyond a threshold value, the module will generate a FIELDDETECTED event. When the strength of the field no longer supports NFC communication, the module will generate a FIELDLOST event. For the Low Power Field Detect threshold values, refer to NFCT Electrical Specification on page 220.

In System OFF, the NFCT Low Power Field Detect function can wake the system up through a reset. The NFC bit in the RESETREAS register in POWER — Power supply on page 60 will be set as the cause of the wake-up.

If the system is put into System OFF mode while a field is already present, the NFCT Low Power Field Detect function will wake the system up right away and generate a reset.

Important: As a consequence of a reset, NFCT is disabled, and therefore the reset handler will have to activate NFCT again and set it up properly.

The HFXO must be running before the NFCT peripheral goes into ACTIVATED state. Note that the NFCT peripheral calibration is automatically done on ACTIVATE task. The HFXO can be turned off when the NFCT peripheral goes into SENSE mode. The shortcut FIELDDETECTED_ACTIVATE can be used when the HFXO is already running while in SENSE mode.

Outgoing data will be collected from RAM with the EasyDMA function and assembled according to theTXD.FRAMECONFIG on page 216 register. Incoming data will be disassembled according to the RXD.FRAMECONFIG register and the data section in the frame will be written to RAM via the EasyDMA function.

The NFCT peripheral includes a frame timing controller that can be used to accurately control the inter-frame delay between the incoming frame and a corresponding outgoing frame. It also includes optional CRC functionality.

6.14.2 Operating states

Tasks and events are used to control the operating state of the peripheral. The module can change state by triggering a task, or when specific operations are finalized. Events and tasks allow software to keep track of and change the current state.

See NFCT block diagram on page 197 and NFCT state diagram, automatic collision resolution enabled on page 199 for more information. See *NFC Forum, NFC Activity Technical Specification* for description on NFCT operating states.

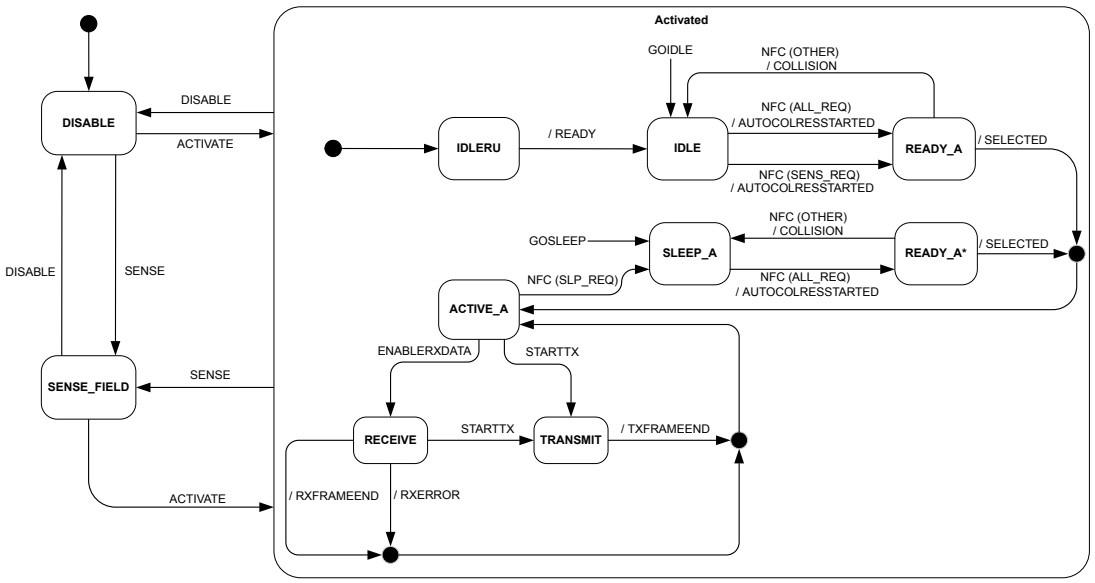

Figure 61: NFCT state diagram, automatic collision resolution enabled

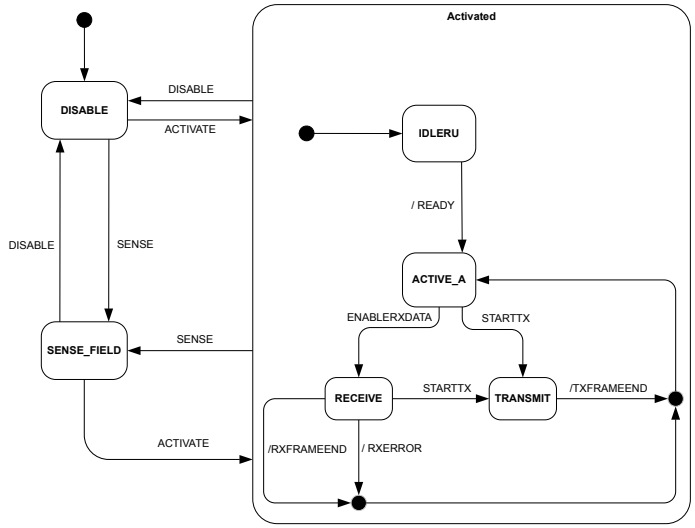

Figure 62: NFCT state diagram, automatic collision resolution disabled

Important:

- FIELDLOST event is not generated in SENSE mode.
- Sending SENSE task while field is still present does not generate FIELDDETECTED event.
- If the FIELDDETECTED event is cleared before sending the ACTIVATE task, then the FIELDDETECTED event shows up again after sending the ACTIVATE task. The shortcut FIELDDETECTED_ACTIVATE can be used to avoid this condition.

6.14.3 Pin configuration

NFCT uses two pins to connect the antenna and these pins are shared with GPIOs.

The PROTECT field in the NFCPINS register in UICR defines the usage of these pins and their protection level against excessive voltages. The content of the NFCPINS register is reloaded at every reset. See Pin assignments on page 524 for the pins used by the NFCT peripheral.

When NFCPINS.PROTECT=NFC, a protection circuit will be enabled on the dedicated pins, preventing the chip from being damaged in the presence of a strong NFC field. The protection circuit will short the two pins together if voltage difference exceeds approximately 2V. The GPIO function on those pins will also be disabled.

When NFCPINS.PROTECT=Disabled, the device will not be protected against strong NFC field damages caught by a connected NFCT antenna, and the NFCT peripheral will not operate as expected, as it will never leave the DISABLE state.

The pins dedicated to the NFCT antenna function will have some limitation when the pins are configured for normal GPIO operation. The pin capacitance will be higher on those (refer to C_{PAD_NFC} in the Electrical Specification of GPIO — General purpose input/output on page 141), and some increased leakage current between the two pins is to be expected if they are used in GPIO mode, and are driven to different logical values. To save power, the two pins should always be set to the same logical value whenever entering one of the device power saving modes. For details, refer to I_{NFC_LEAK} in the Electrical Specification of GPIO — General purpose input/output on page 141.

6.14.4 EasyDMA

The NFCT peripheral implements EasyDMA for reading and writing of data packets from and to the Data RAM.

The NFCT EasyDMA utilizes a pointer called PACKETPTR on page 216 for receiving and transmitting packets.

The NFCT peripheral uses EasyDMA to read or write RAM, but not both at the same time. The event RXFRAMESTART indicates that the EasyDMA has started writing to the RAM for a receive frame and the event RXFRAMEND indicates that the EasyDMA has completed writing to the RAM. Similarly, the event TXFRAMESTART indicates that the EasyDMA has started reading from the RAM for a transmit frame and the event TXFRAMEND indicates that the EasyDMA has completed reading from the RAM. If a transmit and a receive operation is issued at the same time, the transmit operation would be prioritized.

Starting a transmit operation while the EasyDMA is writing a receive frame to the RAM will result in unpredictable behavior. Starting an EasyDMA operation when there is an ongoing EasyDMA operation may result in unpredictable behavior. It is recommended to wait for the TXFRAMEEND or RXFRAMEEND event for the ongoing transmit or receive before starting a new receive or transmit operation.

The MAXLEN on page 216 register determines the maximum number of bytes that can be read from or written to the RAM. This feature can be used to ensure that the NFCT peripheral does not overwrite, or read beyond, the RAM assigned to a packet. Note that if the RXD.AMOUNT or TXD.AMOUNT register indicates longer data packets than set in MAXLEN, the frames sent to or received from the physical layer

will be incomplete. In that situation, in RX, the OVERRUN bit in the FRAMESTATUS.RX register will be set and an RXERROR event will be triggered.

Important: The RXD.AMOUNT and TXD.AMOUNT define a frame length in bytes and bits excluding start of frame (SoF), end of frame (EoF), and parity, but including CRC for RXD.AMOUNT only. Make sure to take potential additional bits into account when setting MAXLEN.

Only sending task ENABLERXDATA ensures that a new value in PACKETPTR pointing to the RX buffer in Data RAM is taken into account.

If PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a hard fault or RAM corruption. For more information about the different memory regions, see Chapter Memory on page 19.

The NFCT peripherals normally do alternative receive and transmit frames. Therefore, to prepare for the next frame, the PACKETPTR, MAXLEN, TXD.FRAMECONFIG and TXD.AMOUNT can be updated while the receive is in progress, and, similarly, the PACKETPTR, MAXLEN and RXD.FRAMECONFIG can be updated while the transmit is in progress. They can be updated and prepared for the next NFC frame immediately after the STARTED event of the current frame has been received. Updating the TXD.FRAMECONFIG and TXD.AMOUNT during the current transmit frame or updating RXD.FRAMECONFIG during current receive frame may cause unpredictable behaviour.

In accordance with *NFC Forum, NFC Digital Protocol Technical Specification*, the least significant bit (LSB) from the least significant byte (LSByte) is sent on air first. The bytes are stored in increasing order, starting at the lowest address in the EasyDMA buffer in RAM.

6.14.5 Frame assembler

The NFCT peripheral implements a frame assembler in hardware.

When the NFCT peripheral is in the ACTIVE_A state, the software can decide to enter RX or TX mode. For RX, see Frame disassembler on page 202. For TX, the software must indicate the address of the source buffer in Data RAM and its size through programming the PACKETPTR and MAXLEN registers respectively, then issuing a STARTTX task.

MAXLEN must be set so that it matches the size of the frame to be sent.

The STARTED event indicates that the PACKETPTR and MAXLEN registers have been captured by the frame assembler EasyDMA.

When asserting the STARTTX task, the frame assembler module will start reading TXD.AMOUNT.TXDATABYTES bytes (plus one additional byte if TXD.AMOUNT.TXDATABITS > 0) from the RAM position set by the PACKETPTR.

The NFCT peripheral transmits the data as read from RAM, adding framing and the CRC calculated on the fly if set in TXD.FRAMECONFIG. The NFCT peripheral will take (8*TXD.AMOUNT.TXDATABYTES + TXD.AMOUNT.TXDATABITS) bits and assemble a frame according to the settings in TXD.FRAMECONFIG. Both short frames, standard frames, and bit-oriented SDD frames as specified in the *NFC Forum, NFC Digital Protocol Technical Specification* can be assembled by the correct setting of the TXD.FRAMECONFIG register.

The bytes will be transmitted on air in the same order as they are read from RAM with a rising bit order within each byte, least significant bit (LSB) first. That is, b0 will be transmitted on air before b1, and so on. The bits read from RAM will be coded into symbols as defined in the *NFC Forum, NFC Digital Protocol Technical Specification*.

Important: Some NFC Forum documents, such as *NFC Forum, NFC Digital Protocol Technical Specification*, define bit numbering in a byte from b1 (LSB) to b8 (most significant bit (MSB)), while most other technical documents from the NFC Forum, and also the Nordic Semiconductor documentation, traditionally number them from b0 to b7. The present document uses the b0–b7 numbering scheme. Be aware of this when comparing the *NFC Forum, NFC Digital Protocol Technical Specification* to others.

The frame assembler can be configured in TXD.FRAMECONFIG to add SoF symbol, calculate and add parity bits, and calculate and add CRC to the data read from RAM when assembling the frame. The total frame will then be longer than what is defined by TXD.AMOUNT.TXDATABYTES. TXDATABITS. DISCARDMODE will select if the first bits in the first byte read from RAM or the last bits in the last byte read from RAM will be discarded if TXD.AMOUNT.TXDATABITS are not equal to zero. Note that if TXD.FRAMECONFIG.PARITY = Parity and TXD.FRAMECONFIG.DISCARDMODE=DiscardStart, a parity bit will be included after the non-complete first byte. No parity will be added after a non-complete last byte.

The frame assemble operation is illustrated in Frame assemble illustration on page 202 for different settings in TXD.FRAMECONFIG. All shaded bit fields are added by the frame assembler. Some of these bits are optional and appearances are configured in TXD.FRAMECONFIG. Note that the frames illustrated do not necessarily comply with the NFC specification. The figure is only to illustrate the behavior of the NFCT peripheral.

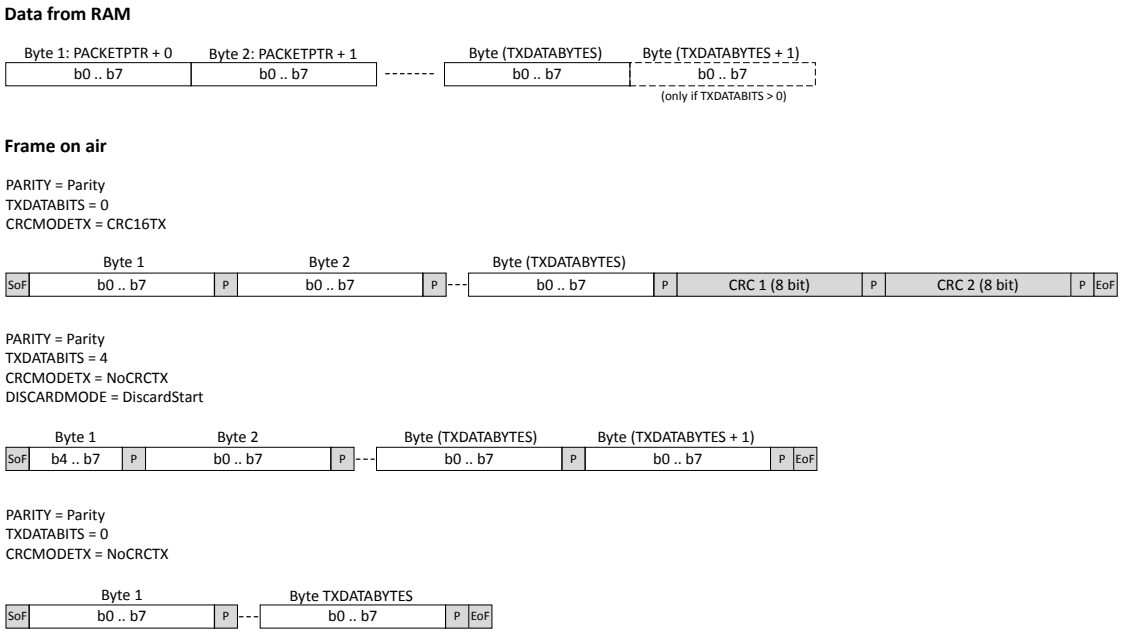

Figure 63: Frame assemble illustration

The accurate timing for transmitting the frame on air is set using the frame timing controller settings.

6.14.6 Frame disassembler

The NFCT peripheral implements a frame disassembler in hardware.

When the NFCT peripheral is in the ACTIVE_A state, the software can decide to enter RX or TX mode. For TX, see Frame assembler on page 201. For RX, the software must indicate the address and size of the destination buffer in Data RAM through programming the PACKETPTR and MAXLEN registers before issuing an ENABLERXDATA task.

The STARTED event indicates that the PACKETPTR and MAXLEN registers have been captured by the frame disassembler EasyDMA.

When an incoming frame starts, the RXFRAMESTART event will get issued and data will be written to the buffer in Data RAM. The frame disassembler will verify and remove any parity bits, start of frame (SoF) and

end of frame (EoF) symbols on the fly based on RXD.FRAMECONFIG register configuration. It will, however, verify and transfer the CRC bytes into RAM, if the CRC is enabled through RXD.FRAMECONFIG.

When an EoF symbol is detected, the NFCT peripheral will assert the RXFRAMEEND event and write the RXD.AMOUNT register to indicate numbers of received bytes and bits in the data packet. The module does not interpret the content of the data received from the remote NFC device, except for SoF, EoF, parity, and CRC checking, as described above. The frame disassemble operation is illustrated below.

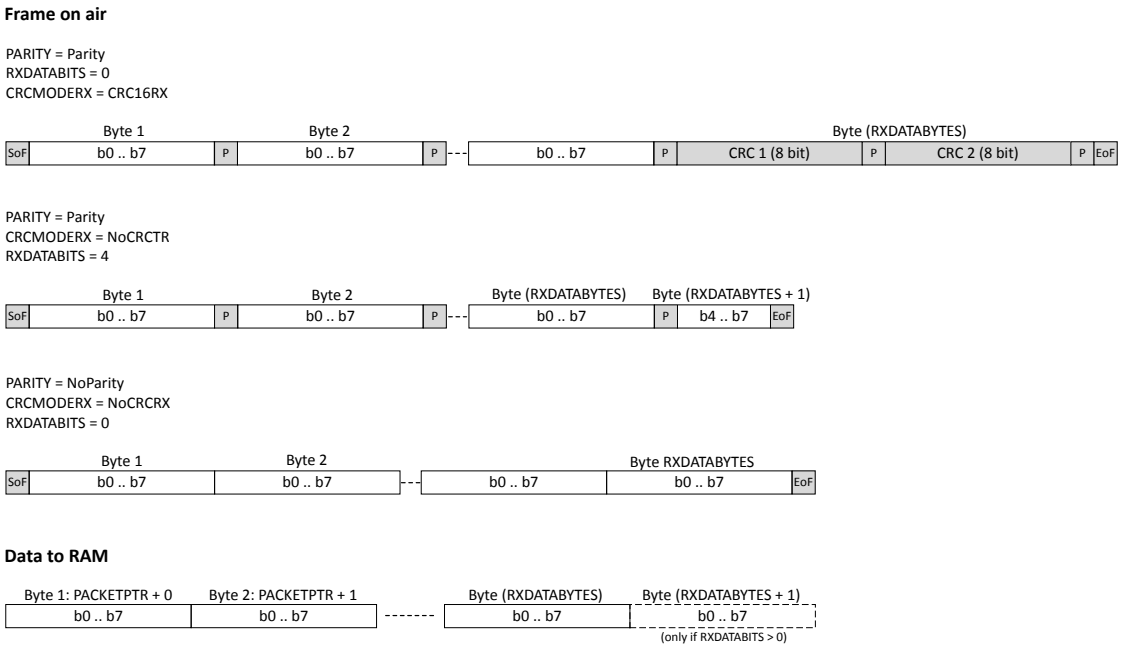

Figure 64: Frame disassemble illustration

Per NFC specification, the time between EoF to the next SoF can be as short as 86 µs, and thefore care must be taken that PACKETPTR and MAXLEN are ready and ENABLERXDATA is issued on time after the end of previous frame. The use of a PPI shortcut from TXFRAMEEND to ENABLERXDATA is recommended.

6.14.7 Frame timing controller

The NFCT peripheral includes a frame timing controller that continuously keeps track of the number of the 13.56 MHz RF carrier clock periods since the end of the EoF of the last received frame.

The NFCT peripheral can be programmed to send a responding frame within a time window or at an exact count of RF carrier periods. In case of FRAMEDELAYMODE = Window, a STARTTX task triggered before the frame timing controller counter is equal to FRAMEDELAYMIN will force the transmission to halt until the counter is equal to FRAMEDELAYMIN. If the counter is within FRAMEDELAYMIN and FRAMEDELAYMAX when the STARTTX task is triggered, the NFCT peripheral will start the transmission straight away. In case of FRAMEDELAYMODE = ExactVal, a STARTTX task triggered before the frame delay counter is equal to FRAMEDELAYMAX will halt the actual transmission start until the counter is equal to FRAMEDELAYMAX.

In case of FRAMEDELAYMODE = WindowGrid, the behaviour is similar to the FRAMEDELAYMODE = Window, but the actual transmission between FRAMEDELAYMIN and FRAMEDELAYMAX starts on a bit grid as defined for NFC-A Listen frames (slot duration of 128 RF carrier periods).

An ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) will be asserted if the frame timing controller counter reaches FRAMEDELAYMAX without any STARTTX task triggered. This may happen even when the response is not required as per *NFC Forum, NFC Digital Protocol Technical Specification*. Any commands handled by the automatic collision resolution that don't involve a response being generated may also result in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS).
The FRAMEDELAYMIN and FRAMEDELAYMAX values shall only be updated before the STARTTX task is triggered. Failing to do so may cause unpredictable behaviour.

The frame timing controller operation is illustrated in Frame timing controller (FRAMEDELAYMODE=Window) on page 204. The frame timing controller automatically adjusts the frame timing counter based on the last received data bit according to NFC-A technology in the *NFC Forum, NFC Digital Protocol Technical Specification*.

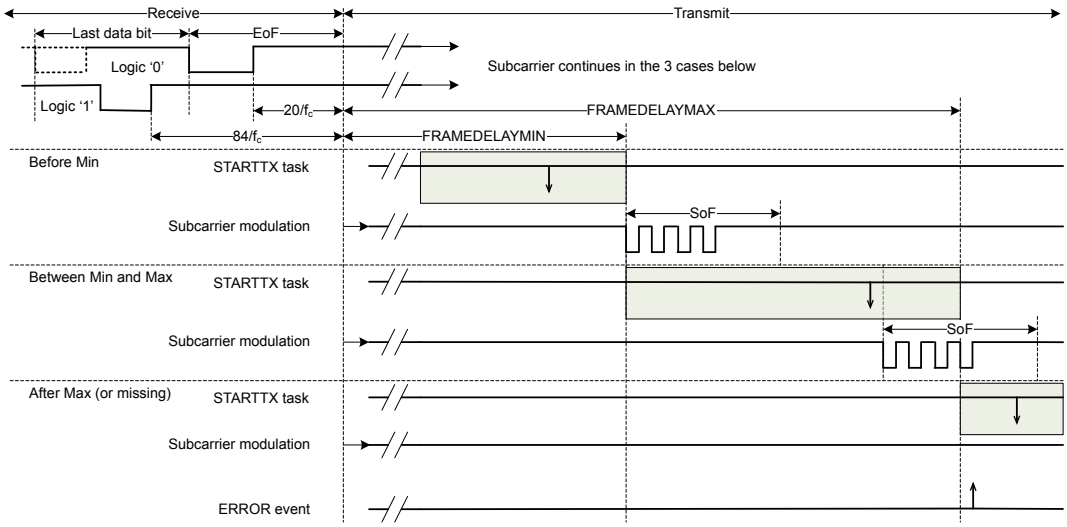

Figure 65: Frame timing controller (FRAMEDELAYMODE=Window)

6.14.8 Collision resolution

The NFCT peripheral implements an automatic collision resolution function as defined by the NFC Forum.

Automatic collision resolution is enabled by default, and it is recommended that the feature is used since it is power efficient and reduces the complexity of software handling the collision resolution sequence. This feature can be disabled through the MODE field in the AUTOCOLRESCONFIG register. When the automatic collision resolution is disabled, all commands will be sent over EasyDMA as defined in frame disassembler.

The SENSRES and SELRES registers need to be programmed upfront in order for the collision resolution to behave correctly. Depending on the NFCIDSIZE field in SENSRES, the following registers also need to be programmed upfront:

- NFCID1_LAST if NFCID1SIZE=NFCID1Single (ID = 4 bytes);
- NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Double (ID = 7 bytes);
- NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Triple (ID = 10 bytes);

A pre-defined set of registers, NFC.TAGHEADER0..3, containing a valid NFCID1 value, is available in FICR and can be used by software to populate the NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST registers.

NFCID1 byte allocation (top sent first on air) on page 205 explains the position of the ID bytes in NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST, depending on the ID size, and as compared to the definition used in the *NFC Forum, NFC Digital Protocol Technical Specification*.

	ID = 4 bytes	ID = 7 bytes	ID = 10 bytes
NFCID1_Q			$nfcid1_0$
NFCID1_R			$nfcid1_1$
NFCID1_S			$nfcid1_2$
NFCID1_T		$nfcid1_0$	$nfcid1_3$
NFCID1_U		$nfcid1_1$	$nfcid1_4$
NFCID1_V		$nfcid1_2$	$nfcid1_5$
NFCID1_W	$nfcid1_0$	$nfcid1_3$	$nfcid1_6$
NFCID1_X	$nfcid1_1$	$nfcid1_4$	$nfcid1_7$
NFCID1_Y	$nfcid1_2$	$nfcid1_5$	$nfcid1_8$
NFCID1_Z	$nfcid1_3$	$nfcid1_6$	$nfcid1_9$

Table 60: NFCID1 byte allocation (top sent first on air)

The hardware implementation can handle the states from IDLE to ACTIVE_A automatically as defined in the *NFC Forum, NFC Activity Technical Specification*, and the other states are to be handled by software. The software keeps track of the state through events. The collision resolution will trigger an AUTOCOLRESSTARTED event when it has started. Reaching the ACTIVE_A state is indicated by the SELECTED event.

If collision resolution fails, a COLLISION event is triggered. Note that errors occurring during automatic collision resolution may also cause ERROR and/or RXERROR events to be generated. Other events may also get generated. It is recommended that the software ignores any event except COLLISION, SELECTED and FIELDLOST during automatic collision resolution. Software shall also make sure that any unwanted SHORT or PPI shortcut is disabled during automatic collision resolution.

The automatic collision resolution will be restarted, if the packets are received with CRC or parity errors while in ACTIVE_A state. The automatic collision resolution feature can be disabled while in ACTIVE_A state to avoid this.

The SLP_REQ is automatically handled by the NFCT peripheral when the automatic collision resolution is enabled. However, this results in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) since the SLP_REQ has no response. This error must be ignored until the SELECTED event is triggered and this error should be cleared by the software when the SELECTED event is triggered.

6.14.9 Antenna interface

In ACTIVATED state, an amplitude regulator will adjust the voltage swing on the antenna pins to a value that is within the V_{swing} limit.

Refer to NFCT Electrical Specification on page 220.

6.14.10 NFCT antenna recommendations

The NFCT antenna coil must be connected differential between **NFC1** and **NFC2** pins of the device.

Two external capacitors should be used to tune the resonance of the antenna circuit to 13.56 MHz.

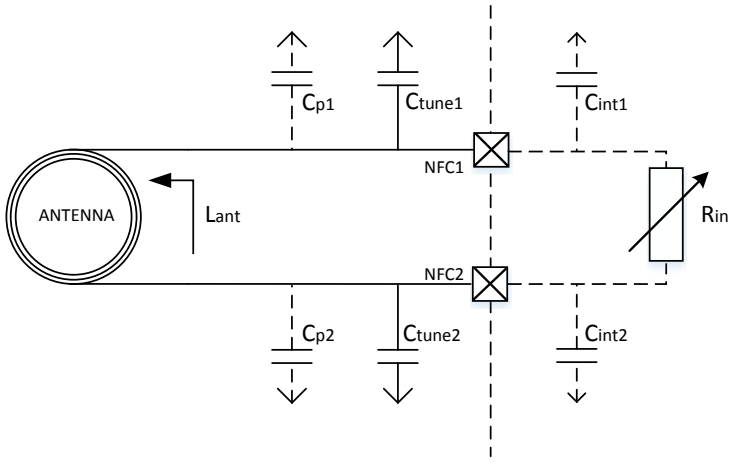

Figure 66: NFCT antenna recommendations

The required tuning capacitor value is given by the below equations:

$$C'_{tune} = \frac{1}{(2\pi \cdot 13.56\ MHz)^2 \cdot L_{ant}} \quad where\ C'_{tune} = \frac{1}{2} \cdot (C_p + C_{\text{int}} + C_{tune})$$

$$and\ C_{tune1} = C_{tune2} = C_{tune} \qquad C_{p1} = C_{p2} = C_p \qquad C_{\text{int1}} = C_{\text{int2}} = C_{\text{int}}$$

$$C_{tune} = \frac{2}{(2\pi \cdot 13.56\ MHz)^2 \cdot L_{ant}} - C_p - C_{\text{int}}$$

An antenna inductance of L_{ant} = 2 µH will give tuning capacitors in the range of 130 pF on each pin. The total capacitance on **NFC1** and **NFC2** must be matched.

6.14.11 Battery protection

If the antenna is exposed to a strong NFC field, current may flow in the opposite direction on the supply due to parasitic diodes and ESD structures.

If the battery used does not tolerate return current, a series diode must be placed between the battery and the device in order to protect the battery.

6.14.12 References

NFC Forum, NFC Analog Specification version 1.0, www.nfc-forum.org

NFC Forum, NFC Digital Protocol Technical Specification version 1.1, www.nfc-forum.org

NFC Forum, NFC Activity Technical Specification version 1.1, www.nfc-forum.org

6.14.13 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40005000	NFCT	NFCT	Near field communication tag	

Table 61: Instances

Register	Offset	Description
TASKS_ACTIVATE	0x000	Activate NFCT peripheral for incoming and outgoing frames, change state to activated
TASKS_DISABLE	0x004	Disable NFCT peripheral
TASKS_SENSE	0x008	Enable NFC sense field mode, change state to sense mode
TASKS_STARTTX	0x00C	Start transmission of an outgoing frame, change state to transmit
TASKS_ENABLERXDATA	0x01C	Initializes the EasyDMA for receive.
TASKS_GOIDLE	0x024	Force state machine to IDLE state
TASKS_GOSLEEP	0x028	Force state machine to SLEEP_A state
EVENTS_READY	0x100	The NFCT peripheral is ready to receive and send frames
EVENTS_FIELDDETECTED	0x104	Remote NFC field detected
EVENTS_FIELDLOST	0x108	Remote NFC field lost
EVENTS_TXFRAMESTART	0x10C	Marks the start of the first symbol of a transmitted frame
EVENTS_TXFRAMEEND	0x110	Marks the end of the last transmitted on-air symbol of a frame
EVENTS_RXFRAMESTART	0x114	Marks the end of the first symbol of a received frame
EVENTS_RXFRAMEEND	0x118	Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer
EVENTS_ERROR	0x11C	NFC error reported. The ERRORSTATUS register contains details on the source of the error.
EVENTS_RXERROR	0x128	NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.
EVENTS_ENDRX	0x12C	RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.
EVENTS_ENDTX	0x130	Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer
EVENTS_AUTOCOLRESSTARTED	0x138	Auto collision resolution process has started
EVENTS_COLLISION	0x148	NFC auto collision resolution error reported.
EVENTS_SELECTED	0x14C	NFC auto collision resolution successfully completed
EVENTS_STARTED	0x150	EasyDMA is ready to receive or send frames.
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSTATUS	0x404	NFC Error Status register
FRAMESTATUS.RX	0x40C	Result of last incoming frame
NFCTAGSTATE	0x410	NfcTag state register
SLEEPSTATE	0x420	Sleep state during automatic collision resolution
FIELDPRESENT	0x43C	Indicates the presence or not of a valid field
FRAMEDELAYMIN	0x504	Minimum frame delay
FRAMEDELAYMAX	0x508	Maximum frame delay
FRAMEDELAYMODE	0x50C	Configuration register for the Frame Delay Timer
PACKETPTR	0x510	Packet pointer for TXD and RXD data storage in Data RAM
MAXLEN	0x514	Size of the RAM buffer allocated to TXD and RXD data storage each
TXD.FRAMECONFIG	0x518	Configuration of outgoing frames
TXD.AMOUNT	0x51C	Size of outgoing frame
RXD.FRAMECONFIG	0x520	Configuration of incoming frames
RXD.AMOUNT	0x524	Size of last incoming frame
NFCID1_LAST	0x590	Last NFCID1 part (4, 7 or 10 bytes ID)
NFCID1_2ND_LAST	0x594	Second last NFCID1 part (7 or 10 bytes ID)
NFCID1_3RD_LAST	0x598	Third last NFCID1 part (10 bytes ID)
AUTOCOLRESCONFIG	0x59C	Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is enabled.
SENSRES	0x5A0	NFC-A SENS_RES auto-response settings
SELRES	0x5A4	NFC-A SEL_RES auto-response settings

Table 62: Register overview

6.14.13.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	FIELDDETECTED_ACTIVATE			Shortcut between FIELDDETECTED event and ACTIVATE task See EVENTS_FIELDDETECTED and TASKS_ACTIVATE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	FIELDLOST_SENSE			Shortcut between FIELDLOST event and SENSE task See EVENTS_FIELDLOST and TASKS_SENSE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	TXFRAMEEND_ENABLERXDATA			Shortcut between TXFRAMEEND event and ENABLERXDATA task See EVENTS_TXFRAMEEND and TASKS_ENABLERXDATA
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.14.13.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY			Enable or disable interrupt for READY event See EVENTS_READY
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	FIELDDETECTED			Enable or disable interrupt for FIELDDETECTED event See EVENTS_FIELDDETECTED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	FIELDLOST			Enable or disable interrupt for FIELDLOST event See EVENTS_FIELDLOST
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	TXFRAMESTART			Enable or disable interrupt for TXFRAMESTART event See EVENTS_TXFRAMESTART
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXFRAMEEND			Enable or disable interrupt for TXFRAMEEND event See EVENTS_TXFRAMEEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	RXFRAMESTART			Enable or disable interrupt for RXFRAMESTART event See EVENTS_RXFRAMESTART
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXFRAMEEND			Enable or disable interrupt for RXFRAMEEND event See EVENTS_RXFRAMEEND
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	ERROR			Enable or disable interrupt for ERROR event See EVENTS_ERROR
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	RXERROR			Enable or disable interrupt for RXERROR event See EVENTS_RXERROR
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	ENDRX			Enable or disable interrupt for ENDRX event See EVENTS_ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	ENDTX			Enable or disable interrupt for ENDTX event See EVENTS_ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	AUTOCOLRESSTARTED			Enable or disable interrupt for AUTOCOLRESSTARTED event See EVENTS_AUTOCOLRESSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	COLLISION			Enable or disable interrupt for COLLISION event See EVENTS_COLLISION
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	SELECTED			Enable or disable interrupt for SELECTED event See EVENTS_SELECTED
			Disabled	0	Disable
			Enabled	1	Enable
T	RW	STARTED			Enable or disable interrupt for STARTED event See EVENTS_STARTED
			Disabled	0	Disable
			Enabled	1	Enable

6.14.13.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FIELDDETECTED			Write '1' to enable interrupt for FIELDDETECTED event See EVENTS_FIELDDETECTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	FIELDLOST			Write '1' to enable interrupt for FIELDLOST event See EVENTS_FIELDLOST
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXFRAMESTART			Write '1' to enable interrupt for TXFRAMESTART event See EVENTS_TXFRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXFRAMEEND			Write '1' to enable interrupt for TXFRAMEEND event See EVENTS_TXFRAMEEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXFRAMESTART			Write '1' to enable interrupt for RXFRAMESTART event See EVENTS_RXFRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXFRAMEEND			Write '1' to enable interrupt for RXFRAMEEND event See EVENTS_RXFRAMEEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	RXERROR			Write '1' to enable interrupt for RXERROR event See EVENTS_RXERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDRX			Write '1' to enable interrupt for ENDRX event See EVENTS_ENDRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDTX			Write '1' to enable interrupt for ENDTX event See EVENTS_ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	AUTOCOLRESSTARTED			Write '1' to enable interrupt for AUTOCOLRESSTARTED event See EVENTS_AUTOCOLRESSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	COLLISION			Write '1' to enable interrupt for COLLISION event See EVENTS_COLLISION
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	SELECTED			Write '1' to enable interrupt for SELECTED event See EVENTS_SELECTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	STARTED			Write '1' to enable interrupt for STARTED event See EVENTS_STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.14.13.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FIELDDETECTED			Write '1' to disable interrupt for FIELDDETECTED event See EVENTS_FIELDDETECTED
			Clear	1	Disable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	FIELDLOST			Write '1' to disable interrupt for FIELDLOST event See EVENTS_FIELDLOST
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXFRAMESTART			Write '1' to disable interrupt for TXFRAMESTART event See EVENTS_TXFRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXFRAMEEND			Write '1' to disable interrupt for TXFRAMEEND event See EVENTS_TXFRAMEEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXFRAMESTART			Write '1' to disable interrupt for RXFRAMESTART event See EVENTS_RXFRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXFRAMEEND			Write '1' to disable interrupt for RXFRAMEEND event See EVENTS_RXFRAMEEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	RXERROR			Write '1' to disable interrupt for RXERROR event See EVENTS_RXERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDRX			Write '1' to disable interrupt for ENDRX event See EVENTS_ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDTX			Write '1' to disable interrupt for ENDTX event See EVENTS_ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
N	RW	AUTOCOLRESSTARTED			Write '1' to disable interrupt for AUTOCOLRESSTARTED event See EVENTS_AUTOCOLRESSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	COLLISION			Write '1' to disable interrupt for COLLISION event See EVENTS_COLLISION
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	SELECTED			Write '1' to disable interrupt for SELECTED event See EVENTS_SELECTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	STARTED			Write '1' to disable interrupt for STARTED event See EVENTS_STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.14.13.5 ERRORSTATUS

Address offset: 0x404

NFC Error Status register

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FRAMEDELAYTIMEOUT			No STARTTX task triggered before expiration of the time set in FRAMEDELAYMAX

6.14.13.6 FRAMESTATUS.RX

Address offset: 0x40C

Result of last incoming frame

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CRCERROR			No valid end of frame (EoF) detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			CRCCorrect	0	Valid CRC detected
			CRCError	1	CRC received does not match local check
B	RW	PARITYSTATUS			Parity status of received frame
			ParityOK	0	Frame received with parity OK
			ParityError	1	Frame received with parity error
C	RW	OVERRUN			Overrun detected
			NoOverrun	0	No overrun detected
			Overrun	1	Overrun error

6.14.13.7 NFCTAGSTATE

Address offset: 0x410

NfcTag state register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	NFCTAGSTATE			NfcTag state
			Disabled	0	Disabled or sense
			RampUp	2	RampUp
			Idle	3	Idle
			Receive	4	Receive
			FrameDelay	5	FrameDelay
			Transmit	6	Transmit

6.14.13.8 SLEEPSTATE

Address offset: 0x420

Sleep state during automatic collision resolution

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	SLEEPSTATE			Reflects the sleep state during automatic collision resolution. Set to IDLE by a GOIDLE task. Set to SLEEP_A when a valid SLEEP_REQ frame is received or by a GOSLEEP task.
			Idle	0	State is IDLE.
			SleepA	1	State is SLEEP_A.

6.14.13.9 FIELDPRESENT

Address offset: 0x43C

Indicates the presence or not of a valid field

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	FIELDPRESENT			Indicates if a valid field is present. Available only in the activated state.
			NoField	0	No valid field detected
			FieldPresent	1	Valid field detected
B	R	LOCKDETECT			Indicates if the low level has locked to the field
			NotLocked	0	Not locked to field
			Locked	1	Locked to field

6.14.13.10 FRAMEDELAYMIN

Address offset: 0x504

Minimum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000480				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMIN			Minimum frame delay in number of 13.56 MHz clocks

6.14.13.11 FRAMEDELAYMAX

Address offset: 0x508

Maximum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00001000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMAX			Maximum frame delay in number of 13.56 MHz clocks

6.14.13.12 FRAMEDELAYMODE

Address offset: 0x50C

Configuration register for the Frame Delay Timer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	RW	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMODE			Configuration register for the Frame Delay Timer
			FreeRun	0	Transmission is independent of frame timer and will start when the STARTTX task is triggered. No timeout.
			Window	1	Frame is transmitted between FRAMEDELAYMIN and FRAMEDELAYMAX
			ExactVal	2	Frame is transmitted exactly at FRAMEDELAYMAX
			WindowGrid	3	Frame is transmitted on a bit grid between FRAMEDELAYMIN and FRAMEDELAYMAX

6.14.13.13 PACKETPTR

Address offset: 0x510

Packet pointer for TXD and RXD data storage in Data RAM

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Packet pointer for TXD and RXD data storage in Data RAM. This address is a byte-aligned RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.14.13.14 MAXLEN

Address offset: 0x514

Size of the RAM buffer allocated to TXD and RXD data storage each

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																								A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXLEN		[0..257]	Size of the RAM buffer allocated to TXD and RXD data storage each

6.14.13.15 TXD.FRAMECONFIG

Address offset: 0x518

Configuration of outgoing frames

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												D		C	B	A
Reset 0x00000017	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**

ID	RW	Field	Value ID	Value	Description
A	RW	PARITY			Indicates if parity is added to the frame
			NoParity	0	Parity is not added to TX frames
			Parity	1	Parity is added to TX frames
B	RW	DISCARDMODE			Discarding unused bits at start or end of a frame
			DiscardEnd	0	Unused bits are discarded at end of frame (EoF)
			DiscardStart	1	Unused bits are discarded at start of frame (SoF)
C	RW	SOF			Adding SoF or not in TX frames
			NoSoF	0	SoF symbol not added
			SoF	1	SoF symbol added
D	RW	CRCMODETX			CRC mode for outgoing frames
			NoCRCTX	0	CRC is not added to the frame
			CRC16TX	1	16 bit CRC added to the frame based on all the data read from RAM that is used in the frame

6.14.13.16 TXD.AMOUNT

Address offset: 0x51C

Size of outgoing frame

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TXDATABITS		[0..7]	Number of bits in the last or first byte read from RAM that shall be included in the frame (excluding parity bit). The DISCARDMODE field in FRAMECONFIG.TX selects if unused bits is discarded at the start or at the end of a frame. A value of 0 data bytes and 0 data bits is invalid.
B	RW	TXDATABYTES		[0..257]	Number of complete bytes that shall be included in the frame, excluding CRC, parity and framing

6.14.13.17 RXD.FRAMECONFIG

Address offset: 0x520

Configuration of incoming frames

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000015				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PARITY			Indicates if parity expected in RX frame
			NoParity	0	Parity is not expected in RX frames
			Parity	1	Parity is expected in RX frames
B	RW	SOF			SoF expected or not in RX frames
			NoSoF	0	SoF symbol is not expected in RX frames
			SoF	1	SoF symbol is expected in RX frames
C	RW	CRCMODERX			CRC mode for incoming frames
			NoCRCRX	0	CRC is not expected in RX frames
			CRC16RX	1	Last 16 bits in RX frame is CRC, CRC is checked and CRCSTATUS updated

6.14.13.18 RXD.AMOUNT

Address offset: 0x524

Size of last incoming frame

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXDATABITS			Number of bits in the last byte in the frame, if less than 8 (including CRC, but excluding parity and SoF/EoF framing). Frames with 0 data bytes and less than 7 data bits are invalid and are not received properly.
B	R	RXDATABYTES			Number of complete bytes received in the frame (including CRC, but excluding parity and SoF/EoF framing)

6.14.13.19 NFCID1_LAST

Address offset: 0x590

Last NFCID1 part (4, 7 or 10 bytes ID)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00006363				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	NFCID1_Z			NFCID1 byte Z (very last byte sent)
B	RW	NFCID1_Y			NFCID1 byte Y
C	RW	NFCID1_X			NFCID1 byte X
D	RW	NFCID1_W			NFCID1 byte W

6.14.13.20 NFCID1_2ND_LAST

Address offset: 0x594

Second last NFCID1 part (7 or 10 bytes ID)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	NFCID1_V			NFCID1 byte V
B	RW	NFCID1_U			NFCID1 byte U
C	RW	NFCID1_T			NFCID1 byte T

6.14.13.21 NFCID1_3RD_LAST

Address offset: 0x598

Third last NFCID1 part (10 bytes ID)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	NFCID1_S			NFCID1 byte S
B	RW	NFCID1_R			NFCID1 byte R
C	RW	NFCID1_Q			NFCID1 byte Q

6.14.13.22 AUTOCOLRESCONFIG

Address offset: 0x59C

Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is enabled.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MODE			Enables/disables auto collision resolution
			Enabled	0	Auto collision resolution enabled
			Disabled	1	Auto collision resolution disabled

6.14.13.23 SENSRES

Address offset: 0x5A0

NFC-A SENS_RES auto-response settings

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E E E D D D D C C B A A A A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	RW	Field	Value ID	Value	Description
A	RW	BITFRAMESDD			Bit frame SDD as defined by the b5:b1 of byte 1 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
			SDD00000	0	SDD pattern 00000
			SDD00001	1	SDD pattern 00001
			SDD00010	2	SDD pattern 00010
			SDD00100	4	SDD pattern 00100
			SDD01000	8	SDD pattern 01000
			SDD10000	16	SDD pattern 10000
B	RW	RFU5			Reserved for future use. Shall be 0.
C	RW	NFCIDSIZE			NFCID1 size. This value is used by the auto collision resolution engine.
			NFCID1Single	0	NFCID1 size: single (4 bytes)
			NFCID1Double	1	NFCID1 size: double (7 bytes)
			NFCID1Triple	2	NFCID1 size: triple (10 bytes)
D	RW	PLATFCONFIG			Tag platform configuration as defined by the b4:b1 of byte 2 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
E	RW	RFU74			Reserved for future use. Shall be 0.

6.14.13.24 SELRES

Address offset: 0x5A4

NFC-A SEL_RES auto-response settings

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	E D D C C B A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	RFU10			Reserved for future use. Shall be 0.
B	RW	CASCADE			Cascade as defined by the b3 of SEL_RES response in the NFC Forum, NFC Digital Protocol Technical Specification (controlled by hardware, shall be 0)
C	RW	RFU43			Reserved for future use. Shall be 0.
D	RW	PROTOCOL			Protocol as defined by the b7:b6 of SEL_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
E	RW	RFU7			Reserved for future use. Shall be 0.

6.14.14 Electrical specification

6.14.14.1 NFCT Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
f_c	Frequency of operation		13.56		MHz
C_{MI}	Carrier modulation index	95			%
DR	Data Rate		106		kbps
V_{sense}	Peak differential Field detect threshold level on NFC1-NFC2[18]		1.2		Vp
I_{max}	Maximum input current on NFCT pins			80	mA

6.14.14.2 NFCT Timing Parameters

Symbol	Description	Min.	Typ.	Max.	Units
$t_{activate}$	Time from task_ACTIVATE in SENSE or DISABLE state to ACTIVATE_A or IDLE state[19]			500	µs
t_{sense}	Time from remote field is present in SENSE mode to FIELDDETECTED event is asserted			20	µs

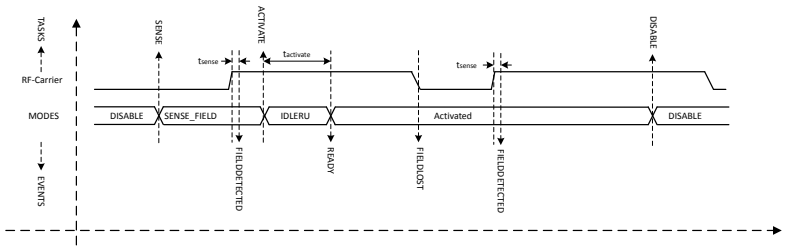

Figure 67: NFCT timing parameters (Shortcuts for FIELDDETECTED and FIELDLOST are disabled)

6.15 PDM — Pulse density modulation interface

The pulse density modulation (PDM) module enables input of pulse density modulated signals from external audio frontends, for example, digital microphones. The PDM module generates the PDM clock

[18] Input is high impedance in sense mode

[19] Does not account for voltage supply and oscillator startup times

and supports single-channel or dual-channel (Left and Right) data input. Data is transferred directly to RAM buffers using EasyDMA.

Listed here are the main features for PDM:

- Up to two PDM microphones configured as a Left/Right pair using the same data input
- 16 kHz output sample rate, 16-bit samples
- EasyDMA support for sample buffering
- HW decimation filters
- Selectable ratio of 64 or 80 between PDM_CLK and output sample rate

The PDM module illustrated in PDM module on page 221 is interfacing up to two digital microphones with the PDM interface. It implements EasyDMA, which relieves real-time requirements associated with controlling the PDM slave from a low priority CPU execution context. It also includes all the necessary digital filter elements to produce PCM samples. The PDM module allows continuous audio streaming.

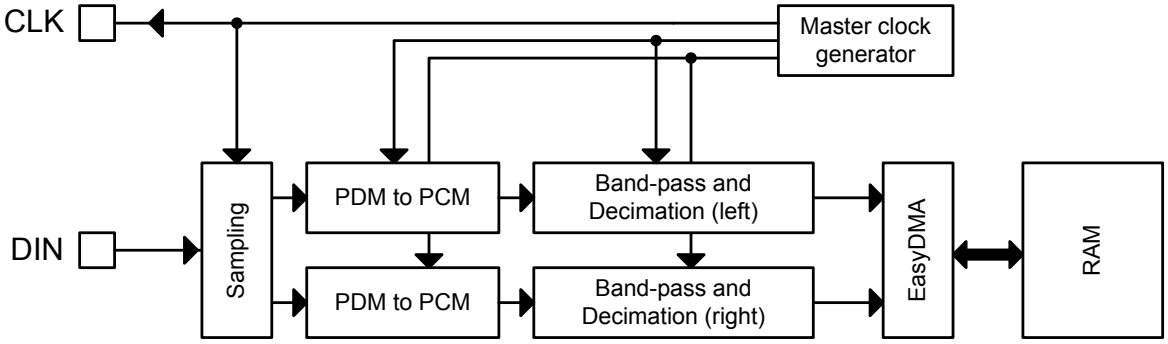

Figure 68: PDM module

6.15.1 Master clock generator

The FREQ field in the master clock's PDMCLKCTRL register allows adjusting the PDM clock's frequency.

The master clock generator does not add any jitter to the HFCLK source chosen. It is recommended (but not mandatory) to use the Xtal as HFCLK source.

6.15.2 Module operation

By default, bits from the left PDM microphone are sampled on PDM_CLK falling edge, bits for the right are sampled on the rising edge of PDM_CLK, resulting in two bitstreams. Each bitstream is fed into a digital filter which converts the PDM stream into 16-bit PCM samples, and filters and down-samples them to reach the appropriate sample rate.

The EDGE field in the MODE register allows swapping Left and Right, so that Left will be sampled on rising edge, and Right on falling.

The PDM module uses EasyDMA to store the samples coming out from the filters into one buffer in RAM.

Depending on the mode chosen in the OPERATION field in the MODE register, memory either contains alternating left and right 16-bit samples (Stereo), or only left 16-bit samples (Mono).

To ensure continuous PDM sampling, it is up to the application to update the EasyDMA destination address pointer as the previous buffer is filled.

The continuous transfer can be started or stopped by sending the START and STOP tasks. STOP becomes effective after the current frame has finished transferring, which will generate the STOPPED event. The STOPPED event indicates that all activity in the module are finished, and that the data is available in RAM (EasyDMA has finished transferring as well). Attempting to restart before receiving the STOPPED event may result in unpredictable behaviour.

6.15.3 Decimation filter

In order to convert the incoming data stream into PCM audio samples, a decimation filter is included in the PDM interface module.

The input of the filter is the two-channel PDM serial stream (with left channel on clock high, right channel on clock low). Depending on the RATIO selected, its output is 2 × 16-bit PCM samples at a sample rate either 64 times or 80 times (depending on the RATIO register) lower than the PDM clock rate.

The filter stage of each channel is followed by a digital volume control, to attenuate or amplify the output samples in a range of -20 dB to +20 dB around the default (reset) setting, defined by $G_{PDM,default}$. The gain is controlled by the GAINL and GAINR registers.

As an example, if the goal is to achieve 2500 RMS output samples (16 bit) with a 1 kHz 90 dBA signal into a -26 dBFS sensitivity PDM microphone, the user will have to sum the PDM module's default gain ($G_{PDM,default}$) and the gain introduced by the microphone and acoustic path of his implementation (an attenuation would translate into a negative gain), and adjust GAINL and GAINR by this amount. Assuming that only the PDM module influences the gain, GAINL and GAINR must be set to $-G_{PDM,default}$ dB to achieve the requirement.

With $G_{PDM,default}$=3.2 dB, and as GAINL and GAINR are expressed in 0.5 dB steps, the closest value to program would be 3.0 dB, which can be calculated as:

```
GAINL = GAINR = (DefaultGain - (2 * 3))
```

Remember to check that the resulting values programmed into GAINL and GAINR fall within MinGain and MaxGain.

6.15.4 EasyDMA

Samples will be written directly to RAM, and EasyDMA must be configured accordingly.

The address pointer for the EasyDMA channel is set in SAMPLE.PTR register. If the destination address set in SAMPLE.PTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

DMA supports Stereo (Left+Right 16-bit samples) and Mono (Left only) data transfer, depending on setting in the OPERATION field in the MODE register. The samples are stored little endian.

MODE.OPERATION	Bits per sample	Result stored per RAM word	Physical RAM allocated (32 bit words)	Result boundary indexes in RAM	Note
Stereo	32 (2x16)	L+R	ceil(SAMPLE.MAXCNT/2)	R0=[31:16]; L0=[15:0]	Default
Mono	16	2xL	ceil(SAMPLE.MAXCNT/2)	L1=[31:16]; L0=[15:0]	

Table 63: DMA sample storage

The destination buffer in RAM consists of one block, the size of which is set in SAMPLE.MAXCNT register. Format is number of 16-bit samples. The physical RAM allocated is always:

```
(RAM allocation, in bytes) = SAMPLE.MAXCNT * 2;
```

(but the mapping of the samples depends on MODE.OPERATION.

If OPERATION=Stereo, RAM will contain a succession of Left and Right samples.

If OPERATION=Mono, RAM will contain a succession of mono samples.

For a given value of SAMPLE.MAXCNT, the buffer in RAM can contain half the stereo sampling time as compared to the mono sampling time.

The PDM acquisition can be started by the START task, after the SAMPLE.PTR and SAMPLE.MAXCNT registers have been written. When starting the module, it will take some time for the filters to start outputting valid data. Transients from the PDM microphone itself may also occur. The first few samples (typically around 50) might hence contain invalid values or transients. It is therefore advised to discard the first few samples after a PDM start.

As soon as the STARTED event is received, the firmware can write the next SAMPLE.PTR value (this register is double-buffered), to ensure continuous operation.

When the buffer in RAM is filled with samples, an END event is triggered. The firmware can start processing the data in the buffer. Meanwhile, the PDM module starts acquiring data into the new buffer pointed to by SAMPLE.PTR, and sends a new STARTED event, so that the firmware can update SAMPLE.PTR to the next buffer address.

6.15.5 Hardware example

Connect the microphone clock to CLK, and data to DIN.

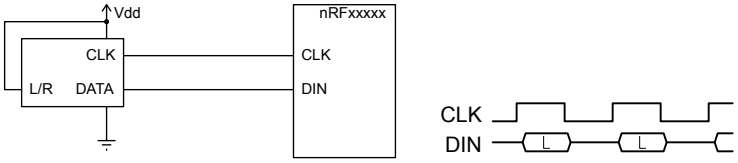

Figure 69: Example of a single PDM microphone, wired as left

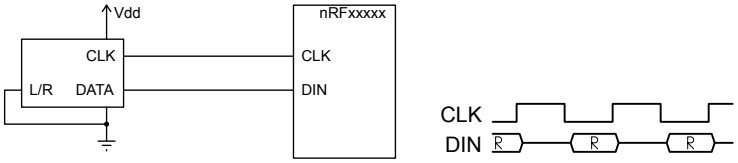

Figure 70: Example of a single PDM microphone, wired as right

Note that in a single-microphone (mono) configuration, depending on the microphone's implementation, either the left or the right channel (sampled at falling or rising CLK edge respectively) will contain reliable data. If two microphones are used, one of them has to be set as left, the other as right (L/R pin tied high or to GND on the respective microphone). It is strongly recommended to use two microphones of exactly the same brand and type so that their timings in left and right operation match.

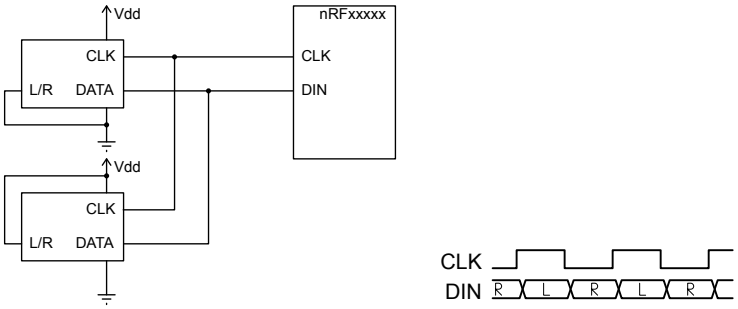

Figure 71: Example of two PDM microphones

6.15.6 Pin configuration

The CLK and DIN signals associated to the PDM module are mapped to physical pins according to the configuration specified in the PSEL.CLK and PSEL.DIN registers respectively. If the CONNECT field in any

PSEL register is set to Disconnected, the associated PDM module signal will not be connected to the required physical pins, and will not operate properly.

The PSEL.CLK and PSEL.DIN registers and their configurations are only used as long as the PDM module is enabled, and retained only as long as the device is in System ON mode. See POWER — Power supply on page 60 for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register.

To ensure correct behaviour in the PDM module, the pins used by the PDM module must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 224 before enabling the PDM module. This is to ensure that the pins used by the PDM module are driven correctly if the PDM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the PDM module is supposed to be connected to an external PDM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behaviour.

PDM signal	PDM pin	Direction	Output value	Comment
CLK	As specified in PSEL.CLK	Output	0	
DIN	As specified in PSEL.DIN	Input	Not applicable	

Table 64: GPIO configuration before enabling peripheral

6.15.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface	

Table 65: Instances

Register	Offset	Description
TASKS_START	0x000	Starts continuous PDM transfer
TASKS_STOP	0x004	Stops PDM transfer
EVENTS_STARTED	0x100	PDM transfer has started
EVENTS_STOPPED	0x104	PDM transfer has finished
EVENTS_END	0x108	The PDM has written the last sample specified by SAMPLE.MAXCNT (or the last sample after a STOP task has been received) to Data RAM
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	PDM module enable register
PDMCLKCTRL	0x504	PDM clock generator control
MODE	0x508	Defines the routing of the connected PDM microphones' signals
GAINL	0x518	Left output gain adjustment
GAINR	0x51C	Right output gain adjustment
RATIO	0x520	Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.
PSEL.CLK	0x540	Pin number configuration for PDM CLK signal
PSEL.DIN	0x544	Pin number configuration for PDM DIN signal
SAMPLE.PTR	0x560	RAM address pointer to write samples to with EasyDMA
SAMPLE.MAXCNT	0x564	Number of samples to allocate memory for in EasyDMA mode

Table 66: Register overview

6.15.7.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Enable or disable interrupt for STARTED event See EVENTS_STARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	END			Enable or disable interrupt for END event See EVENTS_END
			Disabled	0	Disable
			Enabled	1	Enable

6.15.7.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to enable interrupt for STARTED event See EVENTS_STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.15.7.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to disable interrupt for STARTED event See EVENTS_STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to disable interrupt for END event See EVENTS_END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.15.7.4 ENABLE

Address offset: 0x500

PDM module enable register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable PDM module
			Disabled	0	Disable
			Enabled	1	Enable

6.15.7.5 PDMCLKCTRL

Address offset: 0x504

PDM clock generator control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x08400000				**0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQ			PDM_CLK frequency
			1000K	0x08000000	PDM_CLK = 32 MHz / 32 = 1.000 MHz
			Default	0x08400000	PDM_CLK = 32 MHz / 31 = 1.032 MHz. Nominal clock for RATIO=Ratio64.
			1067K	0x08800000	PDM_CLK = 32 MHz / 30 = 1.067 MHz
			1231K	0x09800000	PDM_CLK = 32 MHz / 26 = 1.231 MHz
			1280K	0x0A000000	PDM_CLK = 32 MHz / 25 = 1.280 MHz. Nominal clock for RATIO=Ratio80.
			1333K	0x0A800000	PDM_CLK = 32 MHz / 24 = 1.333 MHz

6.15.7.6 MODE

Address offset: 0x508

Defines the routing of the connected PDM microphones' signals

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OPERATION			Mono or stereo operation
			Stereo	0	Sample and store one pair (Left + Right) of 16bit samples per RAM word R=[31:16]; L=[15:0]
			Mono	1	Sample and store two successive Left samples (16 bit each) per RAM word L1=[31:16]; L0=[15:0]
B	RW	EDGE			Defines on which PDM_CLK edge Left (or mono) is sampled
			LeftFalling	0	Left (or mono) is sampled on falling edge of PDM_CLK
			LeftRising	1	Left (or mono) is sampled on rising edge of PDM_CLK

6.15.7.7 GAINL

Address offset: 0x518

Left output gain adjustment

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000028				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	GAINL			Left output gain adjustment, in 0.5 dB steps, around the default module gain (see electrical parameters) 0x00 -20 dB gain adjust 0x01 -19.5 dB gain adjust (...) 0x27 -0.5 dB gain adjust 0x28 0 dB gain adjust 0x29 +0.5 dB gain adjust (...) 0x4F +19.5 dB gain adjust 0x50 +20 dB gain adjust
			MinGain	0x00	-20dB gain adjustment (minimum)
			DefaultGain	0x28	0dB gain adjustment
			MaxGain	0x50	+20dB gain adjustment (maximum)

6.15.7.8 GAINR

Address offset: 0x51C

Right output gain adjustment

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000028				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	GAINR			Right output gain adjustment, in 0.5 dB steps, around the default module gain (see electrical parameters)
			MinGain	0x00	-20dB gain adjustment (minimum)
			DefaultGain	0x28	0dB gain adjustment
			MaxGain	0x50	+20dB gain adjustment (maximum)

6.15.7.9 RATIO

Address offset: 0x520

Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RATIO			Selects the ratio between PDM_CLK and output sample rate
			Ratio64	0	Ratio of 64
			Ratio80	1	Ratio of 80

6.15.7.10 PSEL.CLK

Address offset: 0x540

Pin number configuration for PDM CLK signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.15.7.11 PSEL.DIN

Address offset: 0x544

Pin number configuration for PDM DIN signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.15.7.12 SAMPLE.PTR

Address offset: 0x560

RAM address pointer to write samples to with EasyDMA

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SAMPLEPTR			Address to write PDM samples to over DMA

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.15.7.13 SAMPLE.MAXCNT

Address offset: 0x564

Number of samples to allocate memory for in EasyDMA mode

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	BUFFSIZE		[0..32767]	Length of DMA RAM allocation in number of samples

6.15.8 Electrical specification

6.15.8.1 PDM Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$f_{PDM,CLK,64}$	PDM clock speed. PDMCLKCTRL = Default (Setting needed for 16MHz sample frequency @ RATIO = Ratio64)		1.032		MHz
$f_{PDM,CLK,80}$	PDM clock speed. PDMCLKCTRL = 1280K (Setting needed for 16MHz sample frequency @ RATIO = Ratio80)		1.280		MHz
$t_{PDM,JITTER}$	Jitter in PDM clock output			20	ns
$T_{dPDM,CLK}$	PDM clock duty cycle	40	50	60	%
$t_{PDM,DATA}$	Decimation filter delay			5	ms
$t_{PDM,cv}$	Allowed clock edge to data valid			125	ns
$t_{PDM,ci}$	Allowed (other) clock edge to data invalid	0			ns
$t_{PDM,s}$	Data setup time at $f_{PDM,CLK}$=1.024 MHz or 1.280 MHz	65			ns
$t_{PDM,h}$	Data hold time at $f_{PDM,CLK}$=1.024 MHz or 1.280 MHz	0			ns
$G_{PDM,default}$	Default (reset) absolute gain of the PDM module		3.2		dB

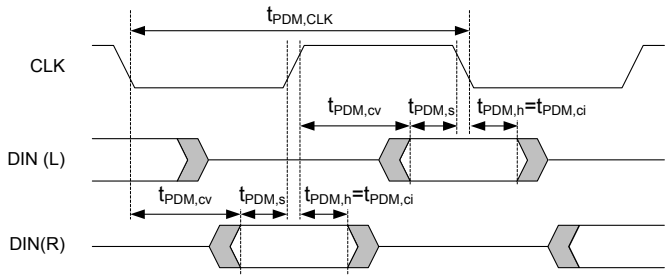

Figure 72: PDM timing diagram

6.16 PPI — Programmable peripheral interconnect

The programmable peripheral interconnect (PPI) enables peripherals to interact autonomously with each other using tasks and events independent of the CPU. The PPI allows precise synchronization between peripherals when real-time application constraints exist and eliminates the need for CPU activity to implement behavior which can be predefined using PPI.

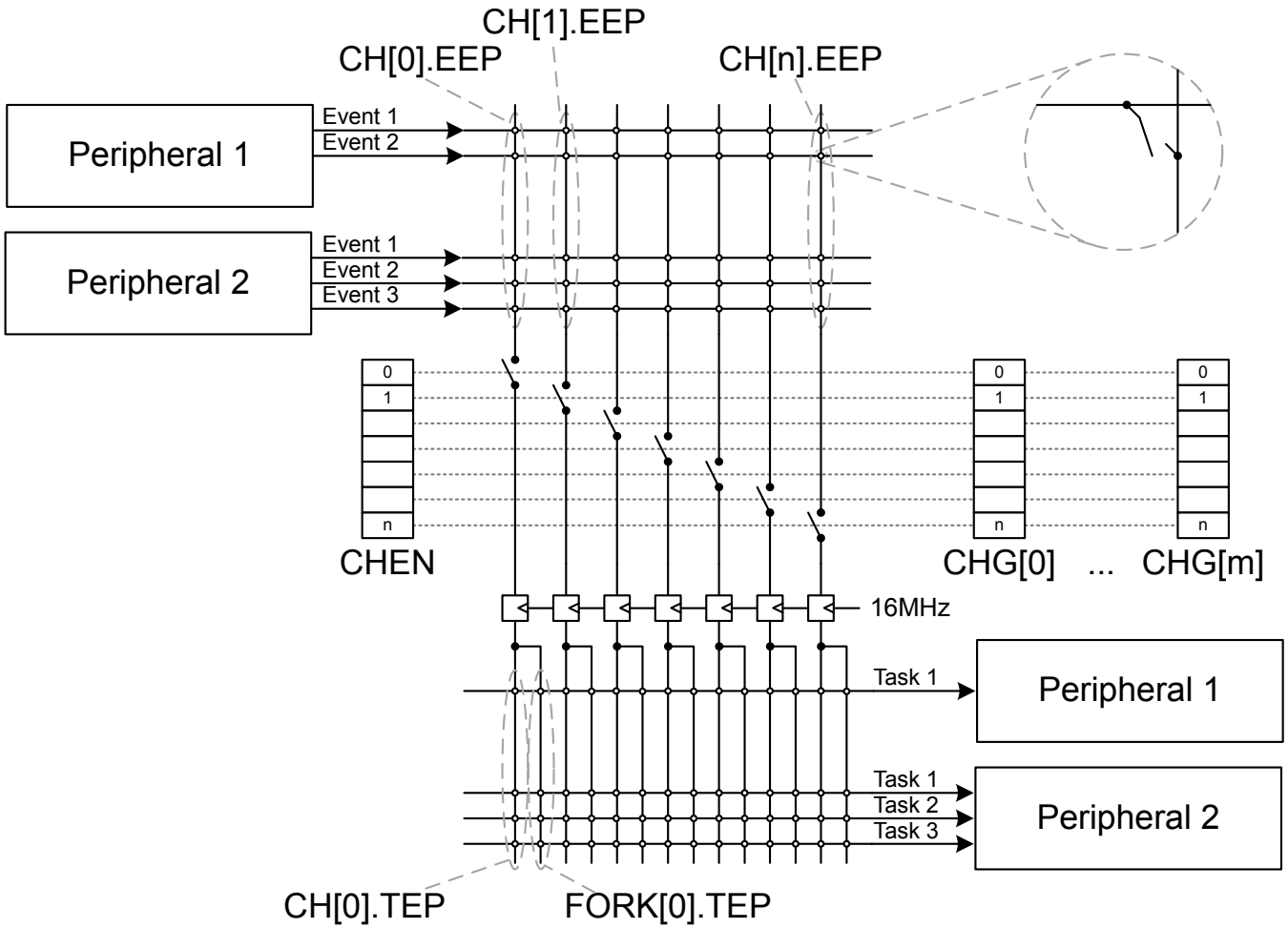

Figure 73: PPI block diagram

The PPI system has, in addition to the fully programmable peripheral interconnections, a set of channels where the event end point (EEP) and task end points (TEP) are fixed in hardware. These fixed channels can be individually enabled, disabled, or added to PPI channel groups (see CHG[n] registers), in the same way as ordinary PPI channels.

Instance	Channel	Number of channels
PPI	0-19	20
PPI (fixed)	20-31	12

Table 67: Configurable and fixed PPI channels

The PPI provides a mechanism to automatically trigger a task in one peripheral as a result of an event occurring in another peripheral. A task is connected to an event through a PPI channel. The PPI channel is composed of three end point registers, one EEP and two TEPs. A peripheral task is connected to a TEP using the address of the task register associated with the task. Similarly, a peripheral event is connected to an EEP using the address of the event register associated with the event.

On each PPI channel, the signals are synchronized to the 16 MHz clock, to avoid any internal violation of setup and hold timings. As a consequence, events that are synchronous to the 16 MHz clock will be delayed by one clock period, while other asynchronous events will be delayed by up to one 16 MHz clock period.

Note that shortcuts (as defined in the SHORTS register in each peripheral) are not affected by this 16 MHz synchronization, and are therefore not delayed.

Each TEP implements a fork mechanism that enables a second task to be triggered at the same time as the task specified in the TEP is triggered. This second task is configured in the task end point register in the FORK registers groups, e.g. FORK.TEP[0] is associated with PPI channel CH[0].

There are two ways of enabling and disabling PPI channels:

- Enable or disable PPI channels individually using the CHEN, CHENSET, and CHENCLR registers.
- Enable or disable PPI channels in PPI channel groups through the groups' ENABLE and DISABLE tasks. Prior to these tasks being triggered, the PPI channel group must be configured to define which PPI channels belong to which groups.

Note that when a channel belongs to two groups m and n, and the tasks CHG[m].EN and CHG[n].DIS occur simultaneously (m and n can be equal or different), the CHG[m].EN on that channel has priority.

PPI tasks (for example, CHG[0].EN) can be triggered through the PPI like any other task, which means they can be hooked to a PPI channel as a TEP. One event can trigger multiple tasks by using multiple channels and one task can be triggered by multiple events in the same way.

6.16.1 Pre-programmed channels

Some of the PPI channels are pre-programmed. These channels cannot be configured by the CPU, but can be added to groups and enabled and disabled like the general purpose PPI channels. The FORK TEP for these channels are still programmable and can be used by the application.

For a list of pre-programmed PPI channels, see the table below.

Channel	EEP	TEP
20	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
21	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
22	TIMER0->EVENTS_COMPARE[1]	RADIO->TASKS_DISABLE
23	RADIO->EVENTS_BCMATCH	AAR->TASKS_START
24	RADIO->EVENTS_READY	CCM->TASKS_KSGEN
25	RADIO->EVENTS_ADDRESS	CCM->TASKS_CRYPT
26	RADIO->EVENTS_ADDRESS	TIMER0->TASKS_CAPTURE[1]
27	RADIO->EVENTS_END	TIMER0->TASKS_CAPTURE[2]
28	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
29	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
30	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_CLEAR
31	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_START

Table 68: Pre-programmed channels

6.16.2 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001F000	PPI	PPI	Programmable peripheral interconnect	

Table 69: Instances

Register	Offset	Description
TASKS_CHG[0].EN	0x000	Enable channel group 0
TASKS_CHG[0].DIS	0x004	Disable channel group 0
TASKS_CHG[1].EN	0x008	Enable channel group 1
TASKS_CHG[1].DIS	0x00C	Disable channel group 1
TASKS_CHG[2].EN	0x010	Enable channel group 2
TASKS_CHG[2].DIS	0x014	Disable channel group 2
TASKS_CHG[3].EN	0x018	Enable channel group 3
TASKS_CHG[3].DIS	0x01C	Disable channel group 3
TASKS_CHG[4].EN	0x020	Enable channel group 4
TASKS_CHG[4].DIS	0x024	Disable channel group 4
TASKS_CHG[5].EN	0x028	Enable channel group 5
TASKS_CHG[5].DIS	0x02C	Disable channel group 5
CHEN	0x500	Channel enable register
CHENSET	0x504	Channel enable set register
CHENCLR	0x508	Channel enable clear register
CH[0].EEP	0x510	Channel 0 event end-point
CH[0].TEP	0x514	Channel 0 task end-point
CH[1].EEP	0x518	Channel 1 event end-point
CH[1].TEP	0x51C	Channel 1 task end-point
CH[2].EEP	0x520	Channel 2 event end-point
CH[2].TEP	0x524	Channel 2 task end-point
CH[3].EEP	0x528	Channel 3 event end-point
CH[3].TEP	0x52C	Channel 3 task end-point
CH[4].EEP	0x530	Channel 4 event end-point
CH[4].TEP	0x534	Channel 4 task end-point
CH[5].EEP	0x538	Channel 5 event end-point
CH[5].TEP	0x53C	Channel 5 task end-point
CH[6].EEP	0x540	Channel 6 event end-point
CH[6].TEP	0x544	Channel 6 task end-point

Register	Offset	Description
CH[7].EEP	0x548	Channel 7 event end-point
CH[7].TEP	0x54C	Channel 7 task end-point
CH[8].EEP	0x550	Channel 8 event end-point
CH[8].TEP	0x554	Channel 8 task end-point
CH[9].EEP	0x558	Channel 9 event end-point
CH[9].TEP	0x55C	Channel 9 task end-point
CH[10].EEP	0x560	Channel 10 event end-point
CH[10].TEP	0x564	Channel 10 task end-point
CH[11].EEP	0x568	Channel 11 event end-point
CH[11].TEP	0x56C	Channel 11 task end-point
CH[12].EEP	0x570	Channel 12 event end-point
CH[12].TEP	0x574	Channel 12 task end-point
CH[13].EEP	0x578	Channel 13 event end-point
CH[13].TEP	0x57C	Channel 13 task end-point
CH[14].EEP	0x580	Channel 14 event end-point
CH[14].TEP	0x584	Channel 14 task end-point
CH[15].EEP	0x588	Channel 15 event end-point
CH[15].TEP	0x58C	Channel 15 task end-point
CH[16].EEP	0x590	Channel 16 event end-point
CH[16].TEP	0x594	Channel 16 task end-point
CH[17].EEP	0x598	Channel 17 event end-point
CH[17].TEP	0x59C	Channel 17 task end-point
CH[18].EEP	0x5A0	Channel 18 event end-point
CH[18].TEP	0x5A4	Channel 18 task end-point
CH[19].EEP	0x5A8	Channel 19 event end-point
CH[19].TEP	0x5AC	Channel 19 task end-point
CHG[0]	0x800	Channel group 0
CHG[1]	0x804	Channel group 1
CHG[2]	0x808	Channel group 2
CHG[3]	0x80C	Channel group 3
CHG[4]	0x810	Channel group 4
CHG[5]	0x814	Channel group 5
FORK[0].TEP	0x910	Channel 0 task end-point
FORK[1].TEP	0x914	Channel 1 task end-point
FORK[2].TEP	0x918	Channel 2 task end-point
FORK[3].TEP	0x91C	Channel 3 task end-point
FORK[4].TEP	0x920	Channel 4 task end-point
FORK[5].TEP	0x924	Channel 5 task end-point
FORK[6].TEP	0x928	Channel 6 task end-point
FORK[7].TEP	0x92C	Channel 7 task end-point
FORK[8].TEP	0x930	Channel 8 task end-point
FORK[9].TEP	0x934	Channel 9 task end-point
FORK[10].TEP	0x938	Channel 10 task end-point
FORK[11].TEP	0x93C	Channel 11 task end-point
FORK[12].TEP	0x940	Channel 12 task end-point
FORK[13].TEP	0x944	Channel 13 task end-point
FORK[14].TEP	0x948	Channel 14 task end-point
FORK[15].TEP	0x94C	Channel 15 task end-point
FORK[16].TEP	0x950	Channel 16 task end-point
FORK[17].TEP	0x954	Channel 17 task end-point
FORK[18].TEP	0x958	Channel 18 task end-point
FORK[19].TEP	0x95C	Channel 19 task end-point
FORK[20].TEP	0x960	Channel 20 task end-point

Register	Offset	Description
FORK[21].TEP	0x964	Channel 21 task end-point
FORK[22].TEP	0x968	Channel 22 task end-point
FORK[23].TEP	0x96C	Channel 23 task end-point
FORK[24].TEP	0x970	Channel 24 task end-point
FORK[25].TEP	0x974	Channel 25 task end-point
FORK[26].TEP	0x978	Channel 26 task end-point
FORK[27].TEP	0x97C	Channel 27 task end-point
FORK[28].TEP	0x980	Channel 28 task end-point
FORK[29].TEP	0x984	Channel 29 task end-point
FORK[30].TEP	0x988	Channel 30 task end-point
FORK[31].TEP	0x98C	Channel 31 task end-point

Table 70: Register overview

6.16.2.1 CHEN

Address offset: 0x500

Channel enable register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Enable or disable channel i
			Disabled	0	Disable channel
			Enabled	1	Enable channel

6.16.2.2 CHENSET

Address offset: 0x504

Channel enable set register

Read: reads value of CH{i} field in CHEN register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Channel i enable set register. Writing '0' has no effect
			Disabled	0	Read: channel disabled
			Enabled	1	Read: channel enabled
			Set	1	Write: Enable channel

6.16.2.3 CHENCLR

Address offset: 0x508

Channel enable clear register

Read: reads value of CH{i} field in CHEN register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Channel i enable clear register. Writing '0' has no effect
			Disabled	0	Read: channel disabled
			Enabled	1	Read: channel enabled
			Clear	1	Write: disable channel

6.16.2.4 CH[n].EEP (n=0..19)

Address offset: 0x510 + (n × 0x8)

Channel n event end-point

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EEP			Pointer to event register. Accepts only addresses to registers from the Event group.

6.16.2.5 CH[n].TEP (n=0..19)

Address offset: 0x514 + (n × 0x8)

Channel n task end-point

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TEP			Pointer to task register. Accepts only addresses to registers from the Task group.

6.16.2.6 CHG[n] (n=0..5)

Address offset: 0x800 + (n × 0x4)

Channel group n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Include or exclude channel i
			Excluded	0	Exclude
			Included	1	Include

6.16.2.7 FORK[n].TEP (n=0..31)

Address offset: 0x910 + (n × 0x4)

Channel n task end-point

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	TEP			Pointer to task register

6.17 PWM — Pulse width modulation

The pulse with modulation (PWM) module enables the generation of pulse width modulated signals on GPIO. The module implements an up or up-and-down counter with four PWM channels that drive assigned GPIOs.

The following are the main features of a PWM module:

- Programmable PWM frequency
- Up to four PWM channels with individual polarity and duty cycle values
- Edge or center-aligned pulses across PWM channels
- Multiple duty cycle arrays (sequences) defined in RAM
- Autonomous and glitch-free update of duty cycle values directly from memory through EasyDMA (no CPU involvement)
- Change of polarity, duty cycle, and base frequency possibly on every PWM period
- RAM sequences can be repeated or connected into loops

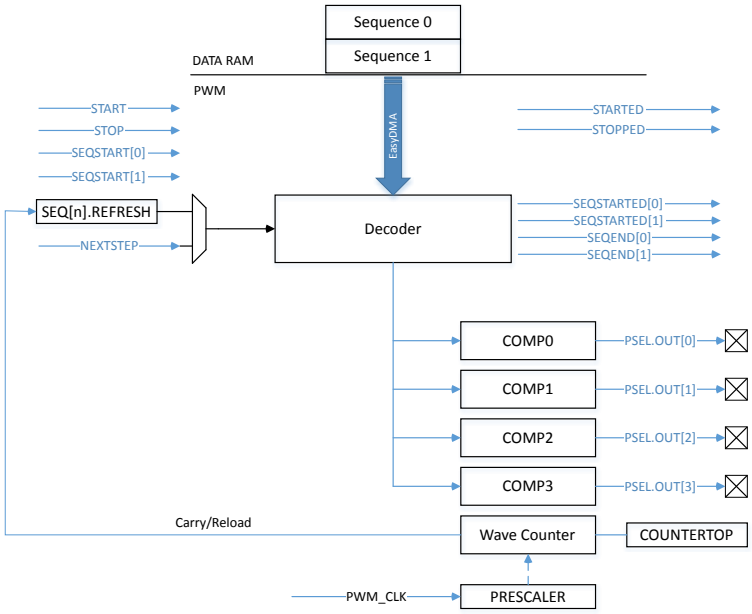

Figure 74: PWM module

6.17.1 Wave counter

The wave counter is responsible for generating the pulses at a duty cycle that depends on the compare values, and at a frequency that depends on COUNTERTOP.

There is one common 15-bit counter with four compare channels. Thus, all four channels will share the same period (PWM frequency), but can have individual duty cycle and polarity. The polarity is set by a value read from RAM (see figure Decoder memory access modes on page 240). Whether the counter counts up, or up and down, is controlled by the MODE register.

The timer top value is controlled by the COUNTERTOP register. This register value, in conjunction with the selected PRESCALER of the PWM_CLK, will result in a given PWM period. A COUNTERTOP value smaller

than the compare setting will result in a state where no PWM edges are generated. OUT[n] is held high, given that the polarity is set to FallingEdge. All compare registers are internal and can only be configured through decoder presented later. COUNTERTOP can be safely written at any time.

Sampling follows the START task. If DECODER.LOAD=WaveForm, the register value is ignored and taken from RAM instead (see section Decoder with EasyDMA on page 240 for more details). If DECODER.LOAD is anything else than the WaveForm, it is sampled following a STARTSEQ[n] task and when loading a new value from RAM during a sequence playback.

The following figure shows the counter operating in up mode (MODE=PWM_MODE_Up), with three PWM channels with the same frequency but different duty cycle:

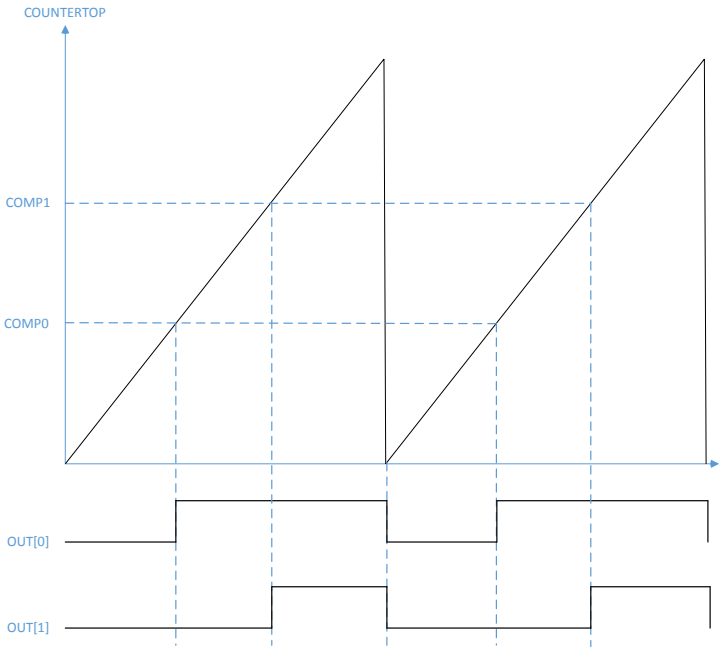

Figure 75: PWM counter in up mode example - FallingEdge polarity

The counter is automatically reset to zero when COUNTERTOP is reached and OUT[n] will invert. OUT[n] is held low if the compare value is 0 and held high if set to COUNTERTOP, given that the polarity is set to

FallingEdge. Counter running in up mode results in pulse widths that are edge-aligned. The following is the code for the counter in up mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up mode, the following formula can be used to compute the PWM period and the step size:

PWM period: $T_{PWM(Up)} = T_{PWM\_CLK} * COUNTERTOP$

Step width/Resolution: $T_{steps} = T_{PWM\_CLK}$

The following figure shows the counter operating in up-and-down mode (MODE=PWM_MODE_UpAndDown), with two PWM channels with the same frequency but different duty cycle and output polarity:

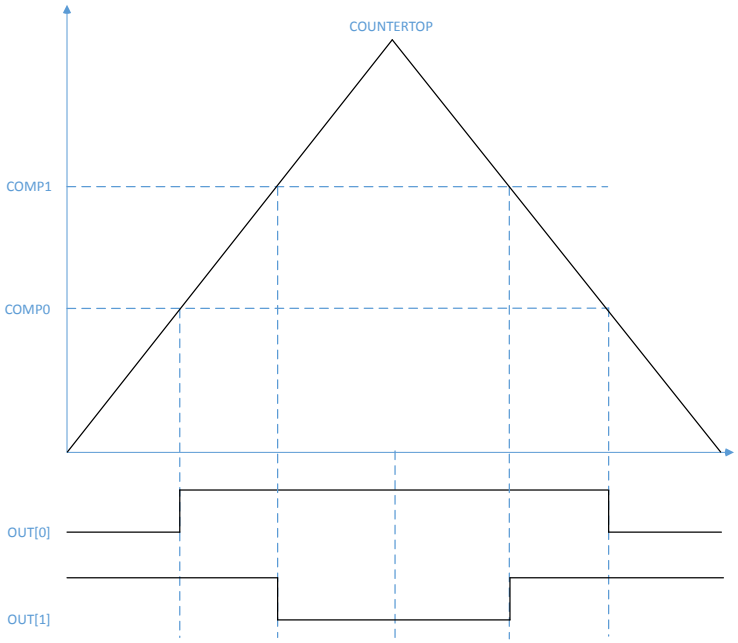

Figure 76: PWM counter in up-and-down mode example

The counter starts decrementing to zero when COUNTERTOP is reached and will invert the OUT[n] when compare value is hit for the second time. This results in a set of pulses that are center-aligned. The following is the code for the counter in up-and-down mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_UpAndDown << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up-and-down mode, the following formula can be used to compute the PWM period and the step size:

$T_{PWM(Up\ And\ Down)} = T_{PWM\_CLK} * 2 * COUNTERTOP$

Step width/Resolution: $T_{steps} = T_{PWM\_CLK} * 2$

6.17.2 Decoder with EasyDMA

The decoder uses EasyDMA to take PWM parameters stored in RAM and update the internal compare registers of the wave counter, based on the mode of operation.

PWM parameters are organized into a sequence containing at least one half word (16 bit). Its most significant bit[15] denotes the polarity of the OUT[n] while bit[14:0] is the 15-bit compare value.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Id				B A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
Id	RW	Field	Value Id	Value	Description
A	RW	COMPARE			Duty cycle setting - value loaded to internal compare register
B	RW	POLARITY			Edge polarity of GPIO.
			RisingEdge	0	First edge within the PWM period is rising
			FallingEdge	1	First edge within the PWM period is falling

The DECODER register controls how the RAM content is interpreted and loaded into the internal compare registers. The LOAD field controls if the RAM values are loaded to all compare channels, or to update a group or all channels with individual values. The following figure illustrates how parameters stored in RAM are organized and routed to various compare channels in different modes:

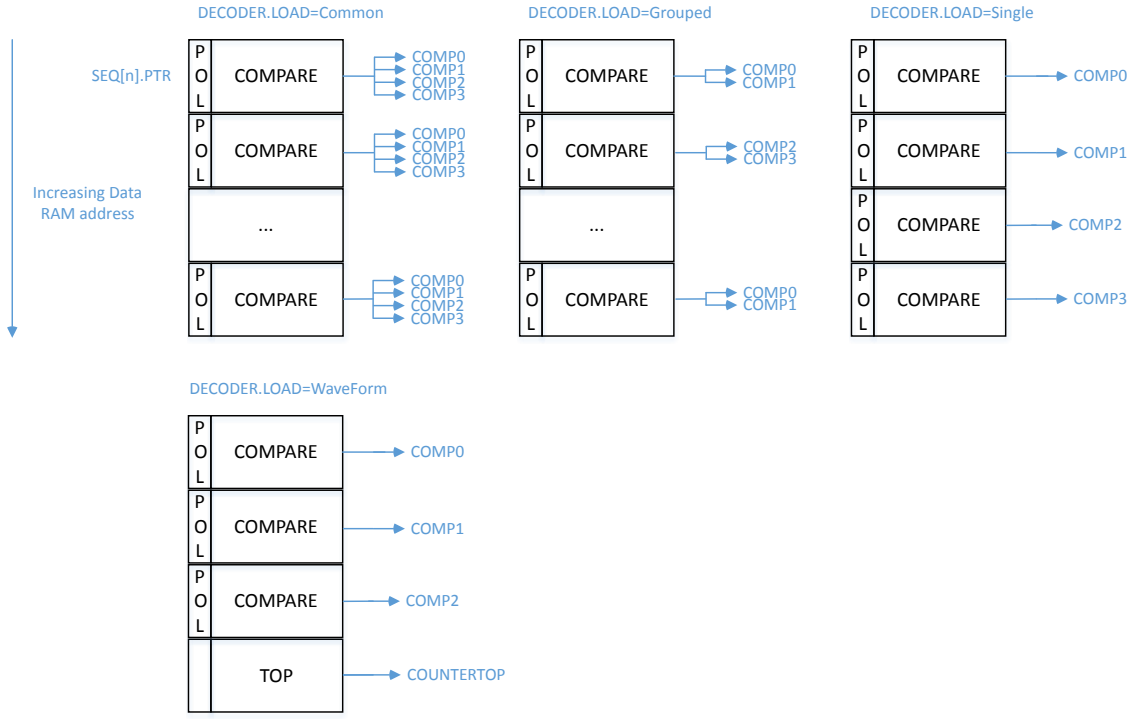

Figure 77: Decoder memory access modes

A special mode of operation is available when DECODER.LOAD is set to WaveForm. In this mode, up to three PWM channels can be enabled - OUT[0] to OUT[2]. In RAM, four values are loaded at a time: the first, second and third location are used to load the values, and the fourth RAM location is used to load the COUNTERTOP register. This way one can have up to three PWM channels with a frequency base that changes on a per PWM period basis. This mode of operation is useful for arbitrary wave form generation in applications, such as LED lighting.

The register SEQ[n].REFRESH=N (one per sequence n=0 or 1) will instruct a new RAM stored pulse width value on every (N+1)th PWM period. Setting the register to zero will result in a new duty cycle update every PWM period, as long as the minimum PWM period is observed.

Note that registers SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored when DECODER.MODE=NextStep. The next value is loaded upon every received NEXTSTEP task.

SEQ[n].PTR is the pointer used to fetch COMPARE values from RAM. If the SEQ[n].PTR is not pointing to a RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions. After the SEQ[n].PTR is set to the desired RAM location, the SEQ[n].CNT register must be set to number of 16-bit half words in the sequence. It is important to observe that the Grouped mode requires one half word per group, while the Single mode requires one half word per channel, thus increasing the RAM size occupation. If PWM generation is not running when the SEQSTART[n] task is triggered, the task will load the first value from RAM and then start the PWM generation. A SEQSTARTED[n] event is generated as soon as the EasyDMA has read the first PWM parameter from RAM and the wave counter has started executing it. When LOOP.CNT=0, sequence n=0 or 1 is played back once. After the last value in the sequence has been loaded and started executing, a SEQEND[n] event is generated. The PWM generation will then continue with the last loaded value. The following figure illustrates an example of such simple playback:

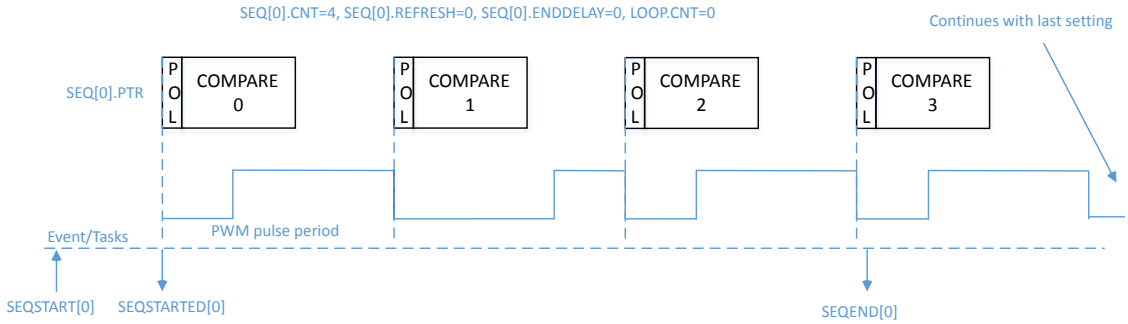

Figure 78: Simple sequence example

Figure depicts the source code used for configuration and timing details in a sequence where only sequence 0 is used and only run once with a new PWM duty cycle for each period.

```
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

To completely stop the PWM generation and force the associated pins to a defined state, a STOP task can be triggered at any time. A STOPPED event is generated when the PWM generation has stopped at the end of currently running PWM period, and the pins go into their idle state as defined in GPIO OUT register. PWM generation can then only be restarted through a SEQSTART[n] task. SEQSTART[n] will resume PWM generation after having loaded the first value from the RAM buffer defined in the SEQ[n].PTR register.

The table below indicates when specific registers get sampled by the hardware. Care should be taken when updating these registers to avoid that values are applied earlier than expected.

Register	Taken into account by hardware	Recommended (safe) update
SEQ[n].PTR	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[n].CNT	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[0].ENDDELAY	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [0] gets loaded from RAM (indicated by the SEQEND[0] event)
		At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].ENDDELAY	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [1] gets loaded from RAM (indicated by the SEQEND[1] event)
		At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
SEQ[0].REFRESH	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].REFRESH	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
COUNTERTOP	In DECODER.LOAD=WaveForm: this register is ignored.	Before starting PWM generation through a SEQSTART[n] task
	In all other LOAD modes: at the end of current PWM period (indicated by the PWMPERIODEND event)	After a STOP task has been triggered, and the STOPPED event has been received.
MODE	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
DECODER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PRESCALER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
LOOP	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PSEL.OUT[n]	Immediately	Before enabling the PWM instance through the ENABLE register

Table 71: When to safely update PWM registers

Note: SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored at the end of a complex sequence, indicated by a LOOPSDONE event. The reason for this is that the last value loaded from RAM is maintained until further action from software (restarting a new sequence, or stopping PWM generation).

A more complex example, where LOOP.CNT>0, is shown in the following figure:

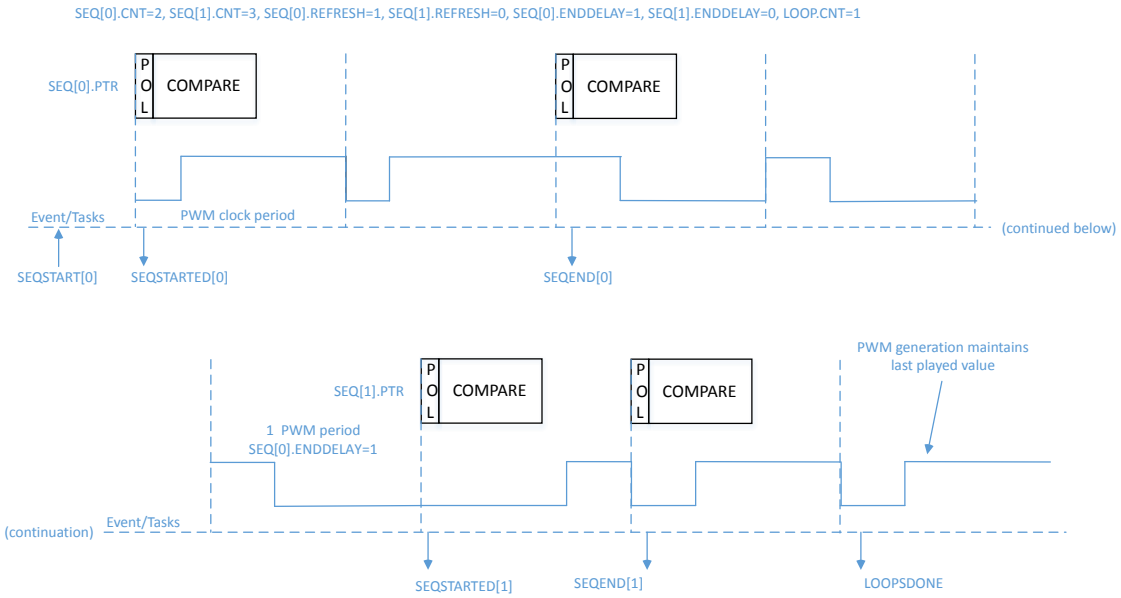

Figure 79: Example using two sequences

In this case, an automated playback takes place, consisting of SEQ[0], delay 0, SEQ[1], delay 1, then again SEQ[0], etc. The user can choose to start a complex playback with SEQ[0] or SEQ[1] through sending the SEQSTART[0] or SEQSTART[1] task. The complex playback always ends with delay 1.

The two sequences 0 and 1 are defined by the addresses of value tables in RAM (pointed to by SEQ[n].PTR) and the buffer size (SEQ[n].CNT). The rate at which a new value is loaded is defined individually for each sequence by SEQ[n].REFRESH. The chaining of sequence 1 following the sequence 0 is implicit, the LOOP.CNT register allows the chaining of sequence 1 to sequence 0 for a determined number of times. In other words, it allows to repeat a complex sequence a number of times in a fully automated way.

In the following code example, sequence 0 is defined with SEQ[0].REFRESH set to 1, meaning that a new PWM duty cycle is pushed every second PWM period. This complex sequence is started with the SEQSTART[0] task, so SEQ[0] is played first. Since SEQ[0].ENDDELAY=1 there will be one PWM period delay between last period on sequence 0 and the first period on sequence 1. Since SEQ[1].ENDDELAY=0 there is no delay 1, so SEQ[0] would be started immediately after the end of SEQ[1]. However, as LOOP.CNT is

1, the playback stops after having played SEQ[1] only once, and both SEQEND[1] and LOOPSDONE are generated (their order is not guaranteed in this case).

```
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                  PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                  PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (1 << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                                                  PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 1;
NRF_PWM0->SEQ[0].ENDDELAY = 1;
NRF_PWM0->SEQ[1].PTR  = ((uint32_t)(seq1_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[1].CNT  = ((sizeof(seq1_ram) / sizeof(uint16_t)) <<
                                                  PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[1].REFRESH  = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

The decoder can also be configured to asynchronously load new PWM duty cycle. If the DECODER.MODE register is set to NextStep, then the NEXTSTEP task will cause an update of internal compare registers on the next PWM period.

The following figures provide an overview of each part of an arbitrary sequence, in various modes (LOOP.CNT=0 and LOOP.CNT>0). In particular, the following are represented:

- Initial and final duty cycle on the PWM output(s)
- Chaining of SEQ[0] and SEQ[1] if LOOP.CNT>0
- Influence of registers on the sequence
- Events generated during a sequence
- DMA activity (loading of next value and applying it to the output(s))

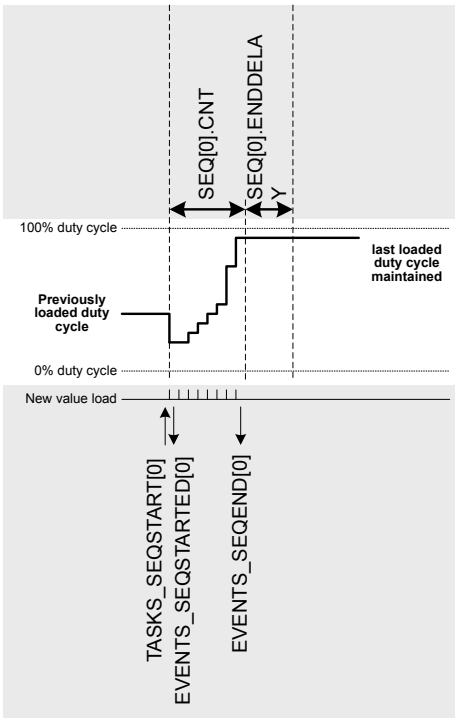

Figure 80: Single shot (LOOP.CNT=0)

Note: The single-shot example also applies to SEQ[1]. Only SEQ[0] is represented for simplicity.

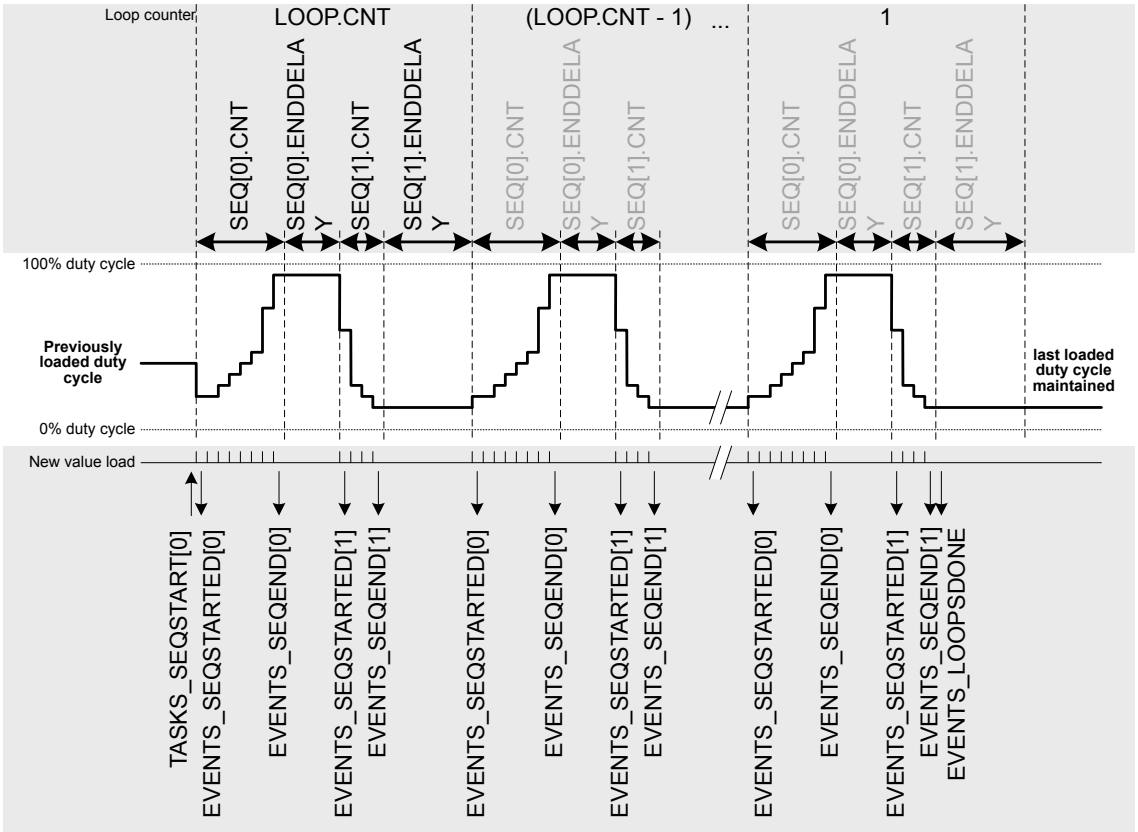

Figure 81: Complex sequence (LOOP.CNT>0) starting with SEQ[0]

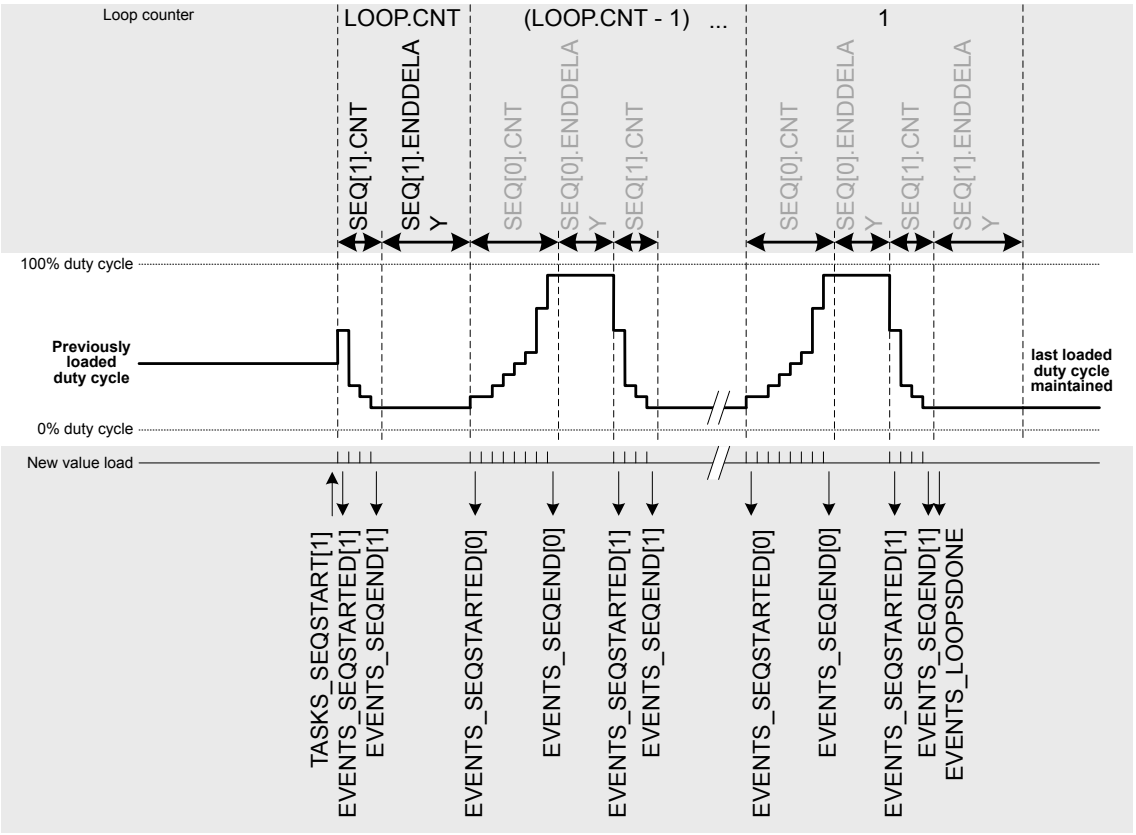

Figure 82: Complex sequence (LOOP.CNT>0) starting with SEQ[1]

Note: If a sequence is in use in a simple or complex sequence, it must have a length of SEQ[n].CNT > 0.

6.17.3 Limitations

Previous compare value is repeated if the PWM period is shorter than the time it takes for the EasyDMA to retrieve from RAM and update the internal compare registers. This is to ensure a glitch-free operation even for very short PWM periods.

6.17.4 Pin configuration

The OUT[n] (n=0..3) signals associated with each PWM channel are mapped to physical pins according to the configuration of PSEL.OUT[n] registers. If PSEL.OUT[n].CONNECT is set to Disconnected, the associated PWM module signal will not be connected to any physical pins.

The PSEL.OUT[n] registers and their configurations are used as long as the PWM module is enabled and the PWM generation active (wave counter started). They are retained only as long as the device is in System ON mode (see section POWER for more information about power modes).

To ensure correct behavior in the PWM module, the pins that are used must be configured in the GPIO peripheral in the following way before the PWM module is enabled:

PWM signal	PWM pin	Direction	Output value	Comment
OUT[n]	As specified in PSEL.OUT[n] (n=0..3)	Output	0	Idle state defined in GPIO OUT register

Table 72: Recommended GPIO configuration before starting PWM generation

The idle state of a pin is defined by the OUT register in the GPIO module, to ensure that the pins used by the PWM module are driven correctly. If PWM generation is stopped by triggering a STOP task, the PWM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected pins (I/Os) for as long as the PWM module is supposed to be connected to an external PWM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

6.17.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001C000	PWM	PWM0	Pulse width modulation unit 0	
0x40021000	PWM	PWM1	Pulse width modulation unit 1	
0x40022000	PWM	PWM2	Pulse width modulation unit 2	
0x4002D000	PWM	PWM3	Pulse width modulation unit 3	

Table 73: Instances

Register	Offset	Description
TASKS_STOP	0x004	Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback
TASKS_SEQSTART[0]	0x008	Loads the first PWM value on all enabled channels from sequence 0, and starts playing that sequence at the rate defined in SEQ[0]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
TASKS_SEQSTART[1]	0x00C	Loads the first PWM value on all enabled channels from sequence 1, and starts playing that sequence at the rate defined in SEQ[1]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
TASKS_NEXTSTEP	0x010	Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.
EVENTS_STOPPED	0x104	Response to STOP task, emitted when PWM pulses are no longer generated
EVENTS_SEQSTARTED[0]	0x108	First PWM period started on sequence 0
EVENTS_SEQSTARTED[1]	0x10C	First PWM period started on sequence 1
EVENTS_SEQEND[0]	0x110	Emitted at end of every sequence 0, when last value from RAM has been applied to wave counter
EVENTS_SEQEND[1]	0x114	Emitted at end of every sequence 1, when last value from RAM has been applied to wave counter
EVENTS_PWMPERIODEND	0x118	Emitted at the end of each PWM period
EVENTS_LOOPSDONE	0x11C	Concatenated sequences have been played the amount of times defined in LOOP.CNT
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	PWM module enable register
MODE	0x504	Selects operating mode of the wave counter
COUNTERTOP	0x508	Value up to which the pulse generator counter counts
PRESCALER	0x50C	Configuration for PWM_CLK
DECODER	0x510	Configuration of the decoder
LOOP	0x514	Number of playbacks of a loop
SEQ[0].PTR	0x520	Beginning address in RAM of this sequence
SEQ[0].CNT	0x524	Number of values (duty cycles) in this sequence
SEQ[0].REFRESH	0x528	Number of additional PWM periods between samples loaded into compare register
SEQ[0].ENDDELAY	0x52C	Time added after the sequence
SEQ[1].PTR	0x540	Beginning address in RAM of this sequence

Register	Offset	Description
SEQ[1].CNT	0x544	Number of values (duty cycles) in this sequence
SEQ[1].REFRESH	0x548	Number of additional PWM periods between samples loaded into compare register
SEQ[1].ENDDELAY	0x54C	Time added after the sequence
PSEL.OUT[0]	0x560	Output pin select for PWM channel 0
PSEL.OUT[1]	0x564	Output pin select for PWM channel 1
PSEL.OUT[2]	0x568	Output pin select for PWM channel 2
PSEL.OUT[3]	0x56C	Output pin select for PWM channel 3

Table 74: Register overview

6.17.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SEQEND0_STOP			Shortcut between SEQEND[0] event and STOP task See EVENTS_SEQEND[0] and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	SEQEND1_STOP			Shortcut between SEQEND[1] event and STOP task See EVENTS_SEQEND[1] and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LOOPSDONE_SEQSTART0			Shortcut between LOOPSDONE event and SEQSTART[0] task See EVENTS_LOOPSDONE and TASKS_SEQSTART[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LOOPSDONE_SEQSTART1			Shortcut between LOOPSDONE event and SEQSTART[1] task See EVENTS_LOOPSDONE and TASKS_SEQSTART[1]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LOOPSDONE_STOP			Shortcut between LOOPSDONE event and STOP task See EVENTS_LOOPSDONE and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.17.5.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
B	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	SEQSTARTED0			Enable or disable interrupt for SEQSTARTED[0] event See EVENTS_SEQSTARTED[0]
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	SEQSTARTED1			Enable or disable interrupt for SEQSTARTED[1] event See EVENTS_SEQSTARTED[1]
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	SEQEND0			Enable or disable interrupt for SEQEND[0] event See EVENTS_SEQEND[0]
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	SEQEND1			Enable or disable interrupt for SEQEND[1] event See EVENTS_SEQEND[1]
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	PWMPERIODEND			Enable or disable interrupt for PWMPERIODEND event See EVENTS_PWMPERIODEND
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	LOOPSDONE			Enable or disable interrupt for LOOPSDONE event See EVENTS_LOOPSDONE
			Disabled	0	Disable
			Enabled	1	Enable

6.17.5.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
B	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	SEQSTARTED0			Write '1' to enable interrupt for SEQSTARTED[0] event See EVENTS_SEQSTARTED[0]
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	SEQSTARTED1			Write '1' to enable interrupt for SEQSTARTED[1] event See EVENTS_SEQSTARTED[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	SEQEND0			Write '1' to enable interrupt for SEQEND[0] event See EVENTS_SEQEND[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SEQEND1			Write '1' to enable interrupt for SEQEND[1] event See EVENTS_SEQEND[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	PWMPERIODEND			Write '1' to enable interrupt for PWMPERIODEND event See EVENTS_PWMPERIODEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	LOOPSDONE			Write '1' to enable interrupt for LOOPSDONE event See EVENTS_LOOPSDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.17.5.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
B	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	SEQSTARTED0			Write '1' to disable interrupt for SEQSTARTED[0] event See EVENTS_SEQSTARTED[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					H G F E D C B
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
D	RW	SEQSTARTED1			Write '1' to disable interrupt for SEQSTARTED[1] event See EVENTS_SEQSTARTED[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	SEQEND0			Write '1' to disable interrupt for SEQEND[0] event See EVENTS_SEQEND[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SEQEND1			Write '1' to disable interrupt for SEQEND[1] event See EVENTS_SEQEND[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	PWMPERIODEND			Write '1' to disable interrupt for PWMPERIODEND event See EVENTS_PWMPERIODEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	LOOPSDONE			Write '1' to disable interrupt for LOOPSDONE event See EVENTS_LOOPSDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.17.5.5 ENABLE

Address offset: 0x500

PWM module enable register

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable PWM module
			Disabled	0	Disabled
			Enabled	1	Enable

6.17.5.6 MODE

Address offset: 0x504

Selects operating mode of the wave counter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	UPDOWN			Selects up mode or up-and-down mode for the counter
			Up	0	Up counter, edge-aligned PWM duty cycle
			UpAndDown	1	Up and down counter, center-aligned PWM duty cycle

6.17.5.7 COUNTERTOP

Address offset: 0x508

Value up to which the pulse generator counter counts

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x000003FF				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	COUNTERTOP		[3..32767]	Value up to which the pulse generator counter counts. This register is ignored when DECODER.MODE=WaveForm and only values from RAM are used.

6.17.5.8 PRESCALER

Address offset: 0x50C

Configuration for PWM_CLK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PRESCALER			Prescaler of PWM_CLK
			DIV_1	0	Divide by 1 (16 MHz)
			DIV_2	1	Divide by 2 (8 MHz)
			DIV_4	2	Divide by 4 (4 MHz)
			DIV_8	3	Divide by 8 (2 MHz)
			DIV_16	4	Divide by 16 (1 MHz)
			DIV_32	5	Divide by 32 (500 kHz)
			DIV_64	6	Divide by 64 (250 kHz)
			DIV_128	7	Divide by 128 (125 kHz)

6.17.5.9 DECODER

Address offset: 0x510

Configuration of the decoder

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LOAD			How a sequence is read from RAM and spread to the compare register
			Common	0	1st half word (16-bit) used in all PWM channels 0..3
			Grouped	1	1st half word (16-bit) used in channel 0..1; 2nd word in channel 2..3
			Individual	2	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in ch.3
			WaveForm	3	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in COUNTERTOP
B	RW	MODE			Selects source for advancing the active sequence
			RefreshCount	0	SEQ[n].REFRESH is used to determine loading internal compare registers
			NextStep	1	NEXTSTEP task causes a new value to be loaded to internal compare registers

6.17.5.10 LOOP

Address offset: 0x514

Number of playbacks of a loop

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CNT			Number of playbacks of pattern cycles
			Disabled	0	Looping disabled (stop at the end of the sequence)

6.17.5.11 SEQ[n].PTR (n=0..1)

Address offset: 0x520 + (n × 0x20)

Beginning address in RAM of this sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Beginning address in RAM of this sequence

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.17.5.12 SEQ[n].CNT (n=0..1)

Address offset: 0x524 + (n × 0x20)

Number of values (duty cycles) in this sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CNT			Number of values (duty cycles) in this sequence
			Disabled	0	Sequence is disabled, and shall not be started as it is empty

6.17.5.13 SEQ[n].REFRESH (n=0..1)

Address offset: 0x528 + (n × 0x20)

Number of additional PWM periods between samples loaded into compare register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	CNT			Number of additional PWM periods between samples loaded into compare register (load every REFRESH.CNT+1 PWM periods)
			Continuous	0	Update every PWM period

6.17.5.14 SEQ[n].ENDDELAY (n=0..1)

Address offset: 0x52C + (n × 0x20)

Time added after the sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CNT			Time added after the sequence in PWM periods

6.17.5.15 PSEL.OUT[n] (n=0..3)

Address offset: 0x560 + (n × 0x4)

Output pin select for PWM channel n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.18 QDEC — Quadrature decoder

The Quadrature decoder (QDEC) provides buffered decoding of quadrature-encoded sensor signals. It is suitable for mechanical and optical sensors.

The sample period and accumulation are configurable to match application requirements. The QDEC provides the following:

- Decoding of digital waveform from off-chip quadrature encoder.
- Sample accumulation eliminating hard real-time requirements to be enforced on application.
- Optional input de-bounce filters.
- Optional LED output signal for optical encoders.

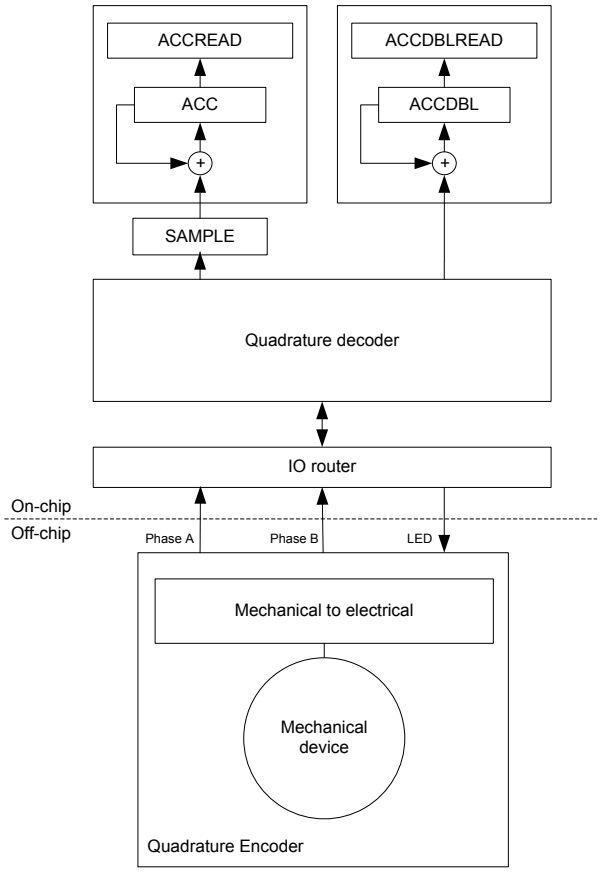

Figure 83: Quadrature decoder configuration

6.18.1 Sampling and decoding

The QDEC decodes the output from an incremental motion encoder by sampling the QDEC phase input pins (A and B).

The off-chip quadrature encoder is an incremental motion encoder outputting two waveforms, phase A and phase B. The two output waveforms are always 90 degrees out of phase, meaning that one always changes level before the other. The direction of movement is indicated by which of these two waveforms that changes level first. Invalid transitions may occur, that is when the two waveforms switch simultaneously. This may occur if the wheel rotates too fast relative to the sample rate set for the decoder.

The QDEC decodes the output from the off-chip encoder by sampling the QDEC phase input pins (A and B) at a fixed rate as specified in the SAMPLEPER register.

If the SAMPLEPER value needs to be changed, the QDEC shall be stopped using the STOP task. SAMPLEPER can be then changed upon receiving the STOPPED event, and QDEC can be restarted using the START task. Failing to do so may result in unpredictable behaviour.

It is good practice to change other registers (LEDPOL, REPORTPER, DBFEN and LEDPRE) only when the QDEC is stopped.

When started, the decoder continuously samples the two input waveforms and decodes these by comparing the current sample pair (n) with the previous sample pair (n-1).

The decoding of the sample pairs is described in the table below.

Previous sample pair(n - 1)		Current samples pair(n)		SAMPLE register	ACC operation	ACCDBL operation	Description
A	B	A	B				
0	0	0	0	0	No change	No change	No movement
0	0	0	1	1	Increment	No change	Movement in positive direction
0	0	1	0	-1	Decrement	No change	Movement in negative direction
0	0	1	1	2	No change	Increment	Error: Double transition
0	1	0	0	-1	Decrement	No change	Movement in negative direction
0	1	0	1	0	No change	No change	No movement
0	1	1	0	2	No change	Increment	Error: Double transition
0	1	1	1	1	Increment	No change	Movement in positive direction
1	0	0	0	1	Increment	No change	Movement in positive direction
1	0	0	1	2	No change	Increment	Error: Double transition
1	0	1	0	0	No change	No change	No movement
1	0	1	1	-1	Decrement	No change	Movement in negative direction
1	1	0	0	2	No change	Increment	Error: Double transition
1	1	0	1	-1	Decrement	No change	Movement in negative direction
1	1	1	0	1	Increment	No change	Movement in positive direction
1	1	1	1	0	No change	No change	No movement

Table 75: Sampled value encoding

6.18.2 LED output

The LED output follows the sample period, and the LED is switched on a given period before sampling and switched off immediately after the inputs are sampled. The period the LED is switched on before sampling is given in the LEDPRE register.

The LED output pin polarity is specified in the LEDPOL register.

For using off-chip mechanical encoders not requiring a LED, the LED output can be disabled by writing value 'Disconnected' to the CONNECT field of the PSEL.LED register. In this case the QDEC will not acquire access to a LED output pin and the pin can be used for other purposes by the CPU.

6.18.3 Debounce filters

Each of the two-phase inputs have digital debounce filters.

When enabled through the DBFEN register, the filter inputs are sampled at a fixed 1 MHz frequency during the entire sample period (which is specified in the SAMPLEPER register), and the filters require all of the samples within this sample period to equal before the input signal is accepted and transferred to the output of the filter.

As a result, only input signal with a steady state longer than twice the period specified in SAMPLEPER are guaranteed to pass through the filter, and any signal with a steady state shorter than SAMPLEPER will always be suppressed by the filter. (This is assumed that the frequency during the debounce period never exceeds 500 kHz (as required by the Nyquist theorem when using a 1 MHz sample frequency).

The LED will always be ON when the debounce filters are enabled, as the inputs in this case will be sampled continuously.

Note that when when the debounce filters are enabled, displacements reported by the QDEC peripheral are delayed by one SAMPLEPER period.

6.18.4 Accumulators

The quadrature decoder contains two accumulator registers, ACC and ACCDBL, that accumulate respectively valid motion sample values and the number of detected invalid samples (double transitions).

The ACC register will accumulate all valid values (1/-1) written to the SAMPLE register. This can be useful for preventing hard real-time requirements from being enforced on the application. When using the ACC register the application does not need to read every single sample from the SAMPLE register, but can instead fetch the ACC register whenever it fits the application. The ACC register will always hold the relative movement of the external mechanical device since the previous clearing of the ACC register. Sample values indicating a double transition (2) will not be accumulated in the ACC register.

An ACCOF event will be generated if the ACC receives a SAMPLE value that would cause the register to overflow or underflow. Any SAMPLE value that would cause an ACC overflow or underflow will be discarded, but any samples not causing the ACC to overflow or underflow will still be accepted.

The accumulator ACCDBL accumulates the number of detected double transitions since the previous clearing of the ACCDBL register.

The ACC and ACCDBL registers can be cleared by the READCLRACC and subsequently read using the ACCREAD and ACCDBLREAD registers.

The ACC register can be separately cleared by the RDCLRACC and subsequently read using the ACCREAD registers.

The ACCDBL register can be separately cleared by the RDCLRDBL and subsequently read using the ACCDBLREAD registers.

The REPORTPER register allows automating the capture of several samples before it can send out a REPORTRDY event in case a non-null displacement has been captured and accumulated, and a DBLRDY event in case one or more double-displacements have been captured and accumulated. The REPORTPER field in this register selects after how many samples the accumulators contents are evaluated to send (or not) REPORTRDY and DBLRDY events.

Using the RDCLRACC task (manually sent upon receiving the event, or using the DBLRDY_RDCLRACC shortcut), ACCREAD can then be read.

In case at least one double transition has been captured and accumulated, a DBLRDY event is sent. Using the RDCLRDBL task (manually sent upon receiving the event, or using the DBLRDY_RDCLRDBL shortcut), ACCDBLREAD can then be read.

6.18.5 Output/input pins

The QDEC uses a three-pin interface to the off-chip quadrature encoder.

These pins will be acquired when the QDEC is enabled in the ENABLE register. The pins acquired by the QDEC cannot be written by the CPU, but they can still be read by the CPU.

The pin numbers to be used for the QDEC are selected using the PSEL.n registers.

6.18.6 Pin configuration

The Phase A, Phase B, and LED signals are mapped to physical pins according to the configuration specified in the PSEL.A, PSEL.B, and PSEL.LED registers respectively.

If the CONNECT field value 'Disconnected' is specified in any of these registers, the associated signal will not be connected to any physical pin. The PSEL.A, PSEL.B, and PSEL.LED registers and their configurations are only used as long as the QDEC is enabled, and retained only as long as the device is in ON mode.

When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register.

To secure correct behavior in the QDEC, the pins used by the QDEC must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 259 before enabling the QDEC. This configuration must be retained in the GPIO for the selected IOs as long as the QDEC is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

QDEC signal	QDEC pin	Direction	Output value	Comment
Phase A	As specified in PSEL.A	Input	Not applicable	
Phase B	As specified in PSEL.B	Input	Not applicable	
LED	As specified in PSEL.LED	Input	Not applicable	

Table 76: GPIO configuration before enabling peripheral

6.18.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40012000	QDEC	QDEC	Quadrature decoder	

Table 77: Instances

Register	Offset	Description
TASKS_START	0x000	Task starting the quadrature decoder
TASKS_STOP	0x004	Task stopping the quadrature decoder
TASKS_READCLRACC	0x008	Read and clear ACC and ACCDBL
TASKS_RDCLRACC	0x00C	Read and clear ACC
TASKS_RDCLRDBL	0x010	Read and clear ACCDBL
EVENTS_SAMPLERDY	0x100	Event being generated for every new sample value written to the SAMPLE register
EVENTS_REPORTRDY	0x104	Non-null report ready
EVENTS_ACCOF	0x108	ACC or ACCDBL register overflow
EVENTS_DBLRDY	0x10C	Double displacement(s) detected
EVENTS_STOPPED	0x110	QDEC has been stopped
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable the quadrature decoder
LEDPOL	0x504	LED output pin polarity
SAMPLEPER	0x508	Sample period
SAMPLE	0x50C	Motion sample value
REPORTPER	0x510	Number of samples to be taken before REPORTRDY and DBLRDY events can be generated
ACC	0x514	Register accumulating the valid transitions
ACCREAD	0x518	Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task
PSEL.LED	0x51C	Pin select for LED signal
PSEL.A	0x520	Pin select for A signal
PSEL.B	0x524	Pin select for B signal
DBFEN	0x528	Enable input debounce filters
LEDPRE	0x540	Time period the LED is switched ON prior to sampling
ACCDBL	0x544	Register accumulating the number of detected double transitions

Register	Offset	Description
ACCDBLREAD	0x548	Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Table 78: Register overview

6.18.7.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																										G	F	E	D	C	B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	REPORTRDY_READCLRACC			Shortcut between REPORTRDY event and READCLRACC task See EVENTS_REPORTRDY and TASKS_READCLRACC
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	SAMPLERDY_STOP			Shortcut between SAMPLERDY event and STOP task See EVENTS_SAMPLERDY and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	REPORTRDY_RDCLRACC			Shortcut between REPORTRDY event and RDCLRACC task See EVENTS_REPORTRDY and TASKS_RDCLRACC
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	REPORTRDY_STOP			Shortcut between REPORTRDY event and STOP task See EVENTS_REPORTRDY and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	DBLRDY_RDCLRDBL			Shortcut between DBLRDY event and RDCLRDBL task See EVENTS_DBLRDY and TASKS_RDCLRDBL
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	DBLRDY_STOP			Shortcut between DBLRDY event and STOP task See EVENTS_DBLRDY and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G	RW	SAMPLERDY_READCLRACC			Shortcut between SAMPLERDY event and READCLRACC task See EVENTS_SAMPLERDY and TASKS_READCLRACC
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.18.7.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SAMPLERDY			Write '1' to enable interrupt for SAMPLERDY event See EVENTS_SAMPLERDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REPORTRDY			Write '1' to enable interrupt for REPORTRDY event See EVENTS_REPORTRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACCOF			Write '1' to enable interrupt for ACCOF event See EVENTS_ACCOF
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	DBLRDY			Write '1' to enable interrupt for DBLRDY event See EVENTS_DBLRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.7.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SAMPLERDY			Write '1' to disable interrupt for SAMPLERDY event See EVENTS_SAMPLERDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REPORTRDY			Write '1' to disable interrupt for REPORTRDY event See EVENTS_REPORTRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACCOF			Write '1' to disable interrupt for ACCOF event See EVENTS_ACCOF

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	DBLRDY			Write '1' to disable interrupt for DBLRDY event See EVENTS_DBLRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.7.4 ENABLE

Address offset: 0x500

Enable the quadrature decoder

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable the quadrature decoder When enabled the decoder pins will be active. When disabled the quadrature decoder pins are not active and can be used as GPIO .
			Disabled	0	Disable
			Enabled	1	Enable

6.18.7.5 LEDPOL

Address offset: 0x504

LED output pin polarity

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LEDPOL			LED output pin polarity
			ActiveLow	0	Led active on output pin low
			ActiveHigh	1	Led active on output pin high

6.18.7.6 SAMPLEPER

Address offset: 0x508

Sample period

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SAMPLEPER			Sample period. The SAMPLE register will be updated for every new sample
			128us	0	128 us
			256us	1	256 us
			512us	2	512 us
			1024us	3	1024 us
			2048us	4	2048 us
			4096us	5	4096 us
			8192us	6	8192 us
			16384us	7	16384 us
			32ms	8	32768 us
			65ms	9	65536 us
			131ms	10	131072 us

6.18.7.7 SAMPLE

Address offset: 0x50C

Motion sample value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	SAMPLE		[-1..2]	Last motion sample The value is a 2's complement value, and the sign gives the direction of the motion. The value '2' indicates a double transition.

6.18.7.8 REPORTPER

Address offset: 0x510

Number of samples to be taken before REPORTRDY and DBLRDY events can be generated

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	REPORTPER			Specifies the number of samples to be accumulated in the ACC register before the REPORTRDY and DBLRDY events can be generated The report period in [us] is given as: RPUS = SP * RP Where RPUS is the report period in [us/report], SP is the sample period in [us/sample] specified in SAMPLEPER, and RP is the report period in [samples/report] specified in REPORTPER .
			10Smpl	0	10 samples / report
			40Smpl	1	40 samples / report
			80Smpl	2	80 samples / report
			120Smpl	3	120 samples / report
			160Smpl	4	160 samples / report
			200Smpl	5	200 samples / report
			240Smpl	6	240 samples / report
			280Smpl	7	280 samples / report
			1Smpl	8	1 sample / report

6.18.7.9 ACC

Address offset: 0x514

Register accumulating the valid transitions

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ACC		[-1024..1023]	Register accumulating all valid samples (not double transition) read from the SAMPLE register Double transitions (SAMPLE = 2) will not be accumulated in this register. The value is a 32 bit 2's complement value. If a sample that would cause this register to overflow or underflow is received, the sample will be ignored and an overflow event (ACCOF) will be generated. The ACC register is cleared by triggering the READCLRACC or the RDCLRACC task.

6.18.7.10 ACCREAD

Address offset: 0x518

Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ACCREAD		[-1024..1023]	Snapshot of the ACC register. The ACCREAD register is updated when the READCLRACC or RDCLRACC task is triggered

6.18.7.11 PSEL.LED

Address offset: 0x51C

Pin select for LED signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.18.7.12 PSEL.A

Address offset: 0x520

Pin select for A signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.18.7.13 PSEL.B

Address offset: 0x524

Pin select for B signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.18.7.14 DBFEN

Address offset: 0x528

Enable input debounce filters

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DBFEN			Enable input debounce filters
			Disabled	0	Debounce input filters disabled
			Enabled	1	Debounce input filters enabled

6.18.7.15 LEDPRE

Address offset: 0x540

Time period the LED is switched ON prior to sampling

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A
Reset 0x00000010				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LEDPRE		[1..511]	Period in us the LED is switched on prior to sampling

6.18.7.16 ACCDBL

Address offset: 0x544

Register accumulating the number of detected double transitions

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ACCDBL		[0..15]	Register accumulating the number of detected double or illegal transitions. (SAMPLE = 2). When this register has reached its maximum value the accumulation of double / illegal transitions will stop. An overflow event (ACCOF) will be generated if any double or illegal transitions are detected after the maximum value was reached. This field is cleared by triggering the READCLRACC or RDCLRDBL task.

6.18.7.17 ACCDBLREAD

Address offset: 0x548

Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ACCDBLREAD		[0..15]	Snapshot of the ACCDBL register. This field is updated when the READCLRACC or RDCLRDBL task is triggered.

6.18.8 Electrical specification

6.18.8.1 QDEC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{SAMPLE}	Time between sampling signals from quadrature decoder	128		131072	µs
t_{LED}	Time from LED is turned on to signals are sampled	0		511	µs

6.19 QSPI — Quad serial peripheral interface

The QSPI peripheral provides support for communicating with an external flash memory device using SPI.

Listed here are the main features for the QSPI peripheral:

- Single/dual/quad SPI input/output
- 2–32 MHz configurable clock frequency
- Single-word read/write access from/to external flash
- EasyDMA for block read and write transfers
- Execute in place (XIP) for executing program directly from external flash
- XIP execution speed: Up to 8 million instruction fetches per second for 16 bit instructions, up to 4 million instruction fetches per second for 32 bit instructions.

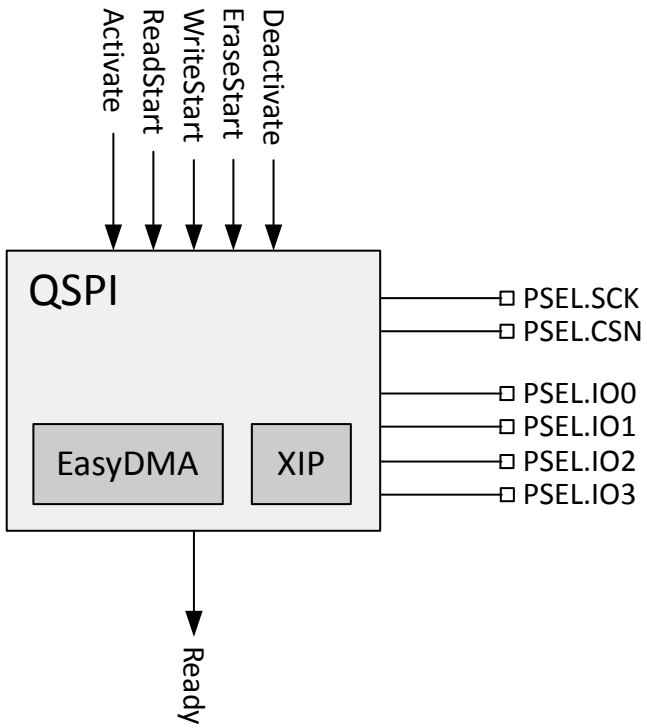

Figure 84: Block diagram

6.19.1 Configuring peripheral

Before any data can be transferred to or from the external flash memory, the peripheral needs to be configured.

1. Select input/output pins in PSEL.SCK on page 279, PSEL.CSN on page 279, PSEL.IO0 on page 279, PSEL.IO1 on page 280, PSEL.IO2 on page 280, and PSEL.IO3 on page 280. See Reference circuitry on page 528 for the recommended pins.

2. To ensure stable operation, set the GPIO drive strength to “high drive”. See the GPIO — General purpose input/output on page 141 chapter for details on how to configure GPIO drive strength.
3. Configure the interface towards the external flash memory using IFCONFIG0 on page 281, IFCONFIG1 on page 282, and ADDRCONF on page 283.
4. Enable the QSPI peripheral and acquire I/O pins using ENABLE on page 277.
5. Activate the external flash memory interface using the ACTIVATE task. The READY event will be generated when the interface has been activated and the external flash memory is ready for access.

Important: If the IFCONFIG0 on page 281 register is configured to use the quad mode, the external flash device also needs to be set in the quad mode before any data transfers can take place.

This can be done by sending custom instructions to the external flash device, as described in Sending custom instructions on page 269.

6.19.2 Write operation

A write operation to the external flash is configured using the WRITE.DST on page 278, WRITE.SRC on page 278, and WRITE.CNT on page 278 registers and started using the WRITESTART task.

The READY event is generated when the transfer is complete.

The QSPI peripheral automatically takes care of splitting DMA transfers into page writes.

6.19.3 Read operation

A read operation from the external flash is configured using the READ.SRC on page 277, READ.DST on page 277, and READ.CNT on page 277 registers and started using the READSTART task.

The READY event is generated when the transfer is complete.

6.19.4 Erase operation

Erase of pages/blocks of the external flash is configured using the ERASE.PTR on page 278 and ERASE.LEN on page 279 registers and started using the ERASESTART task.

The READY event is generated when the erase operation has been started.

Note that in this case the READY event will not indicate that the erase operation of the flash has been completed, but it only signals that the erase operation has been started. The actual status of the erase operation can normally be read from the external flash using a custom instruction, see Sending custom instructions on page 269.

6.19.5 Execute in place

Execute in place (XIP) allows the CPU to execute program code directly from the external flash.

After the external flash has been configured, the CPU can execute code from the external flash by accessing the XIP memory region. See the figure below and Memory map on page 20 for details.

When accessing the XIP memory region, the start address of this XIP memory region will map to the address XIPOFFSET on page 281 of the external flash.

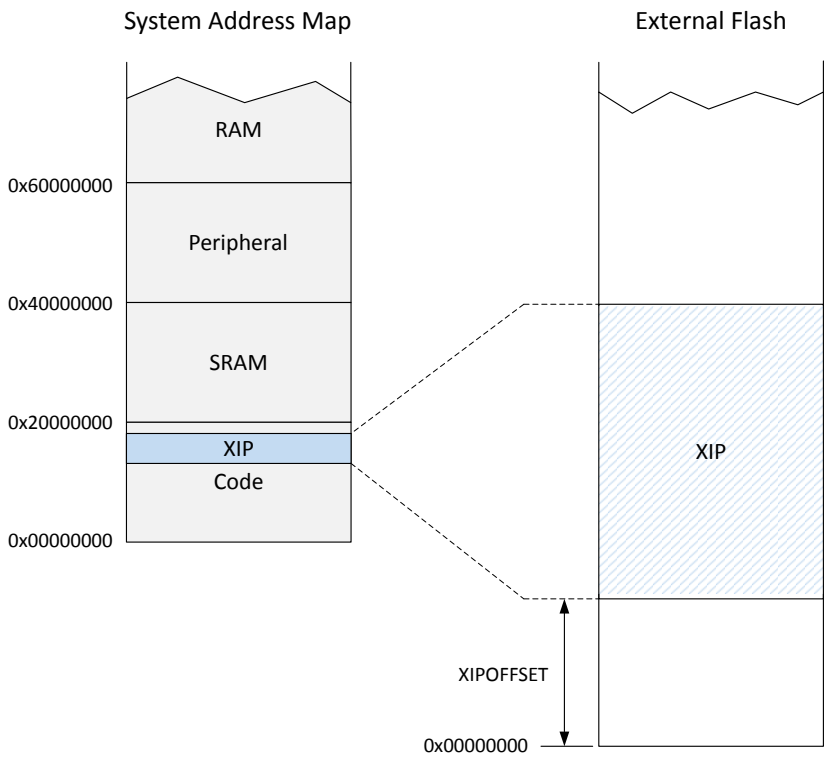

Figure 85: XIP memory map

6.19.6 Sending custom instructions

Custom instructions can be sent to the external flash using the CINSTRCONF on page 283, CINSTRDAT0 on page 284, and CINSTRDAT1 on page 284 registers. It is possible to send an instruction consisting of a one-byte opcode and up to 8 bytes of additional data and to read its response.

A custom instruction is prepared by first writing the data to be sent to CINSTRDAT0 on page 284 and CINSTRDAT1 on page 284 before writing the opcode and other configurations to the CINSTRCONF on page 283 register.

The custom instruction is sent when the CINSTRCONF on page 283 register is written and it is always sent on a single data line SPI interface.

The READY event will be generated when the custom instruction has been sent.

After a custom instruction has been sent, the CINSTRDAT0 on page 284 and CINSTRDAT1 on page 284 will contain the response bytes from the custom instruction.

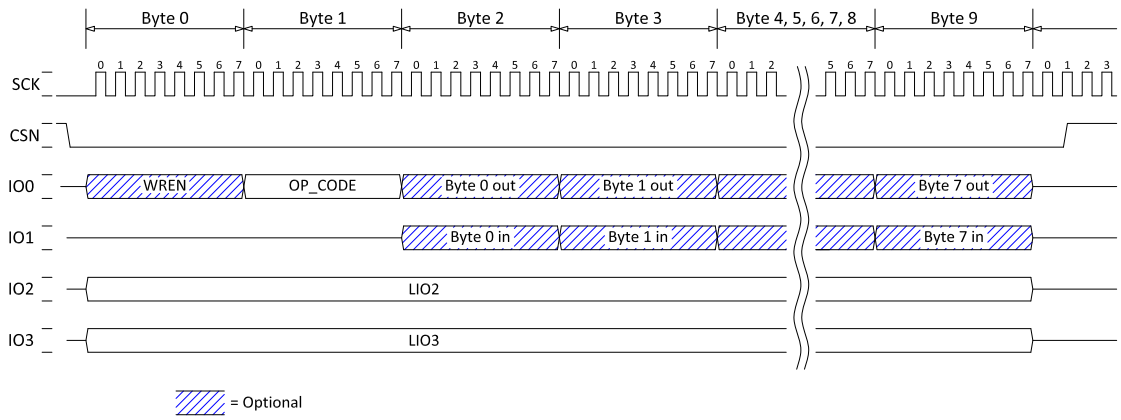

Figure 86: Sending custom instruction

6.19.6.1 Long frame mode

The LFEN and LFSTOP fields in the CINSTRCONF on page 283 control the operation of the custom instruction long frame mode. The long frame mode is a mechanism that permits arbitrary byte length custom instructions. While in long frame mode a long custom instruction sequence is split in multiple writes to the CINSTRDAT0 on page 284 and CINSTRDAT1 on page 284 registers.

To enable the long frame mode every write to the CINSTRCONF on page 283 register must have the LFEN field set to 1. The contents of the OPCODE field will be transmitted after the first write to CINSTRCONF on page 283 and will be omitted in every subsequent write to this register. For subsequent writes the number of data bytes as specified in the LENGTH field are transferred (that is the value of LENGTH - 1 data bytes). The values of the LIO2 and LIO3 fields are set in the first write to CINSTRCONF on page 283 and will apply for the entire custom instruction transmission until the long frame is finalized.

To finalize a long frame transmission, the LFSTOP field in CINSTRCONF on page 283 must be set to 1 in the last write to this register.

6.19.7 Deep power-down mode

The external flash memory can be put in deep power-down mode (DPM) to minimize its current consumption when there is no need to access the memory.

DPM is enabled in the IFCONFIG0 on page 281 register and configured in the DPMDUR on page 283 register. The DPM status of the external memory can be read in the STATUS on page 282 register. The DPMDUR register has to be configured according to the external flash specification to get the information in the STATUS register and the timing of the READY event correct.

Entering/exiting DPM is controlled using the IFCONFIG1 on page 282 register.

6.19.8 Instruction set

The table below shows the instruction set being used by the QSPI peripheral when communicating with an external flash device.

Instruction	Opcode	Description
WREN	0x06	Write enable
RDSR	0x05	Read status register
WRSR	0x01	Write status register
FASTREAD	0x0B	Read bytes at higher speed
READ2O	0x3B	Dual-read output
READ2IO	0xBB	Dual-read input/output
READ4O	0x6B	Quad-read output
READ4IO	0xEB	Quad-read input/output
PP	0x02	Page program
PP2O	0xA2	Dual-page program output
PP4O	0x32	Quad-page program output
PP4IO	0x38	Quad-page program input/output
SE	0x20	Sector erase
BE	0xD8	Block erase
CE	0xC7	Chip erase
DP	0xB9	Enter deep power-down mode
DPE	0xAB	Exit deep power-down mode
EN4B	Specified in the ADDRCONF on page 283 register	Enable 32 bit address mode

Table 79: Instruction set

6.19.9 Interface description

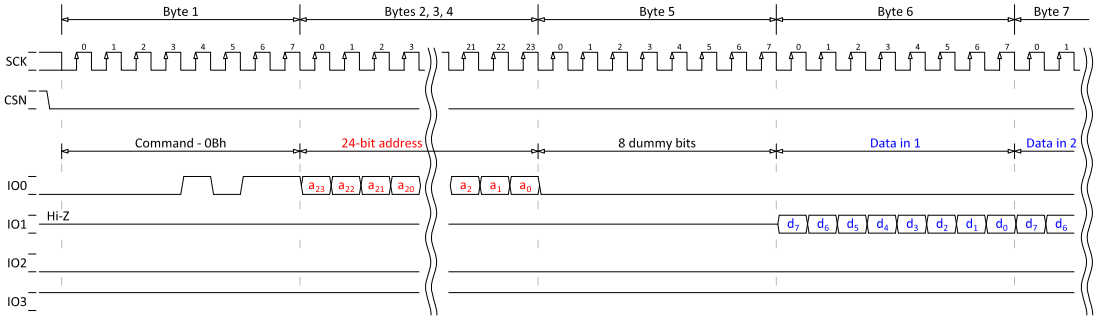

Figure 87: 24-bit FASTREAD, SPIMODE = MODE0

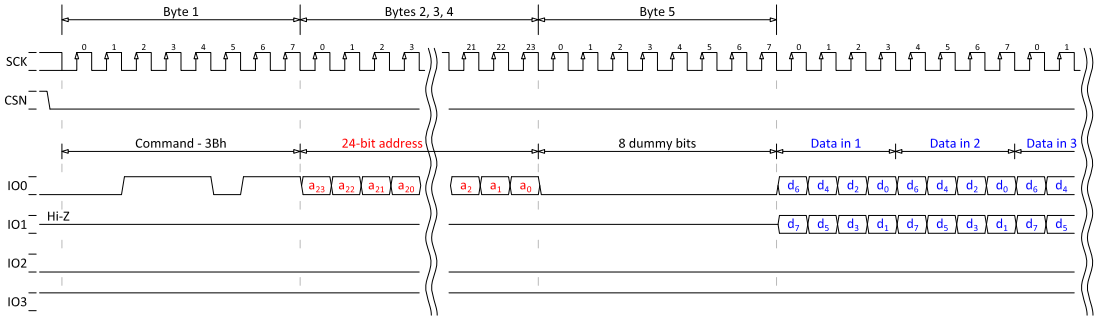

Figure 88: 24-bit READ2O (dual-read output), SPIMODE = MODE0

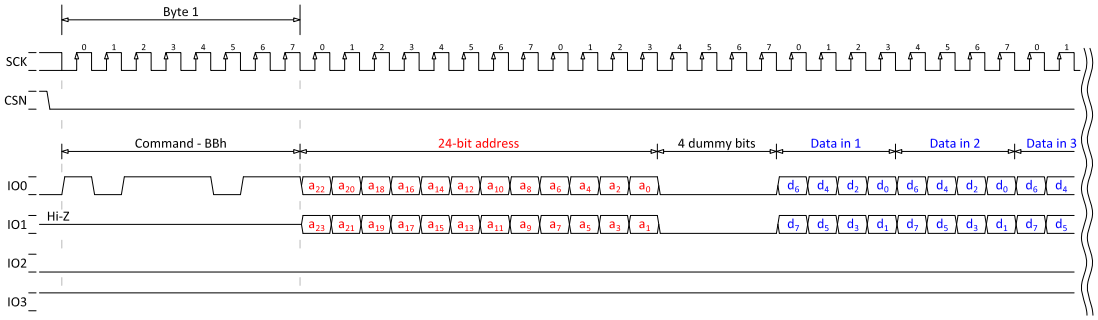

Figure 89: 24-bit READ2IO (dual read input/output), SPIMODE = MODE0

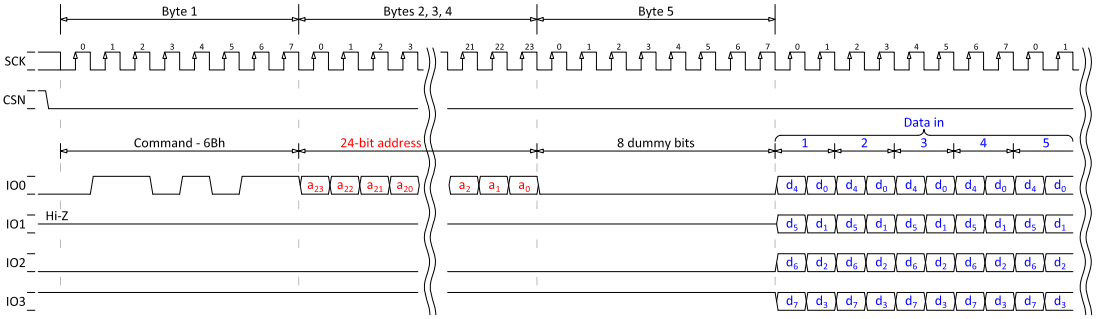

Figure 90: 24-bit READ4O (quad-read output), SPIMODE = MODE0

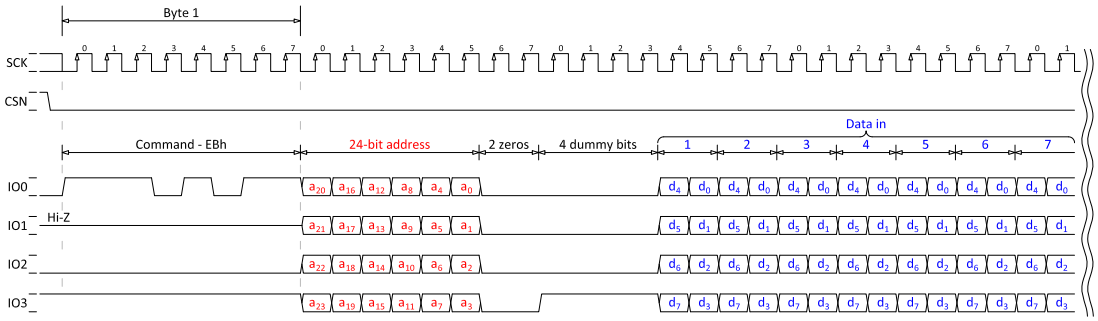

Figure 91: 24-bit READ4IO (quad-read input/output), SPIMODE = MODE0

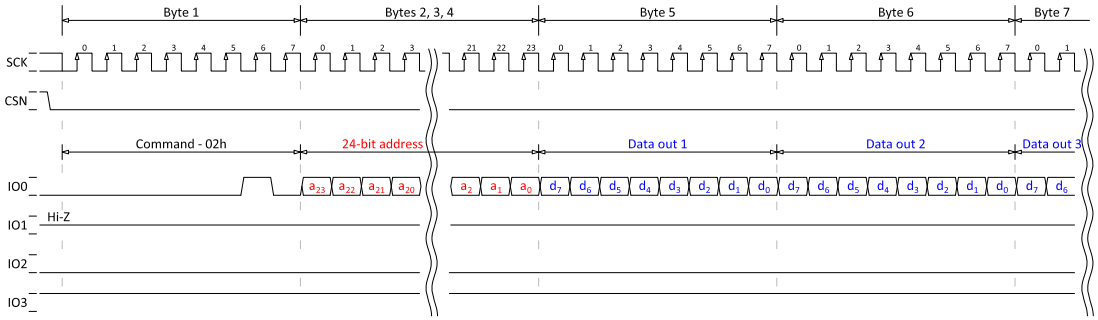

Figure 92: 24-bit PP (page program), SPIMODE = MODE0

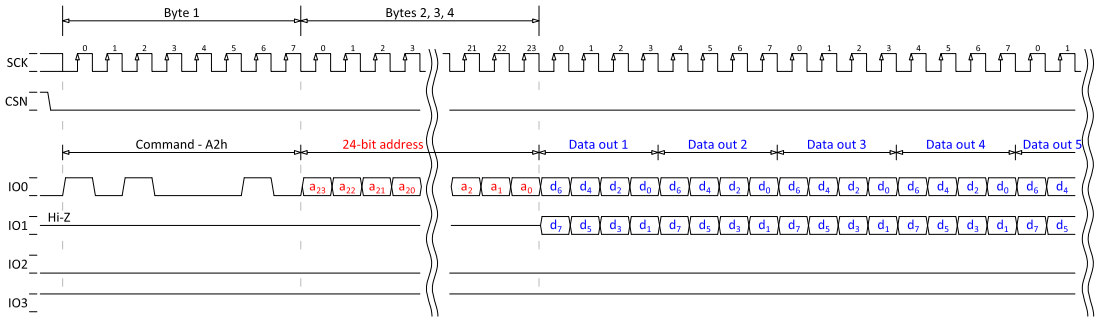

Figure 93: 24-bit PP2O (dual-page program output), SPIMODE = MODE0

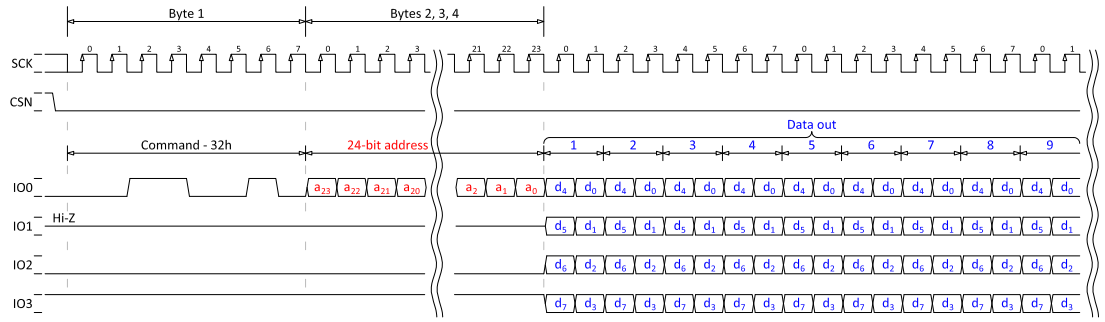

Figure 94: 24-bit PP4O (quad page program output), SPIMODE = MODE0

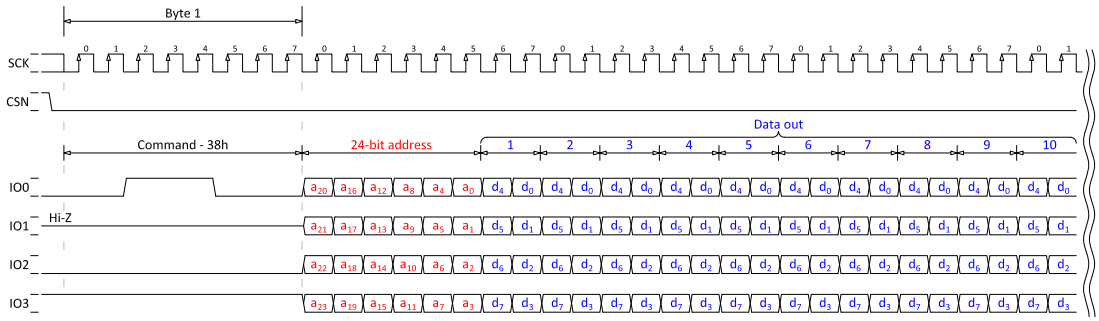

Figure 95: 24-bit PP4IO (quad page program input/output), SPIMODE = MODE0

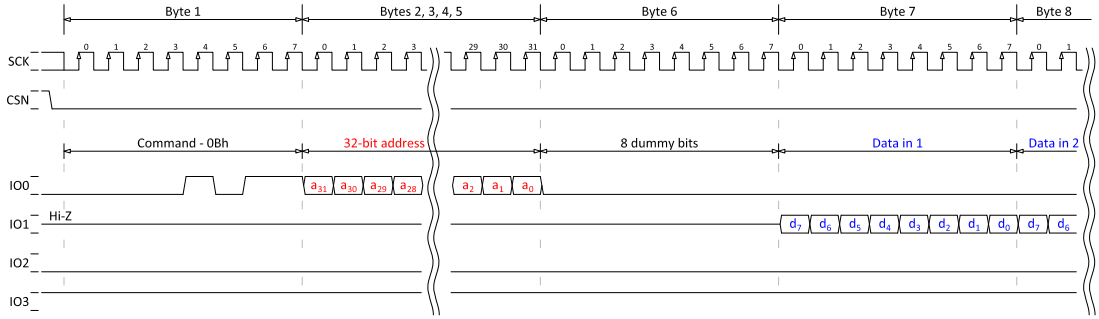

Figure 96: 32-bit FASTREAD, SPIMODE = MODE0

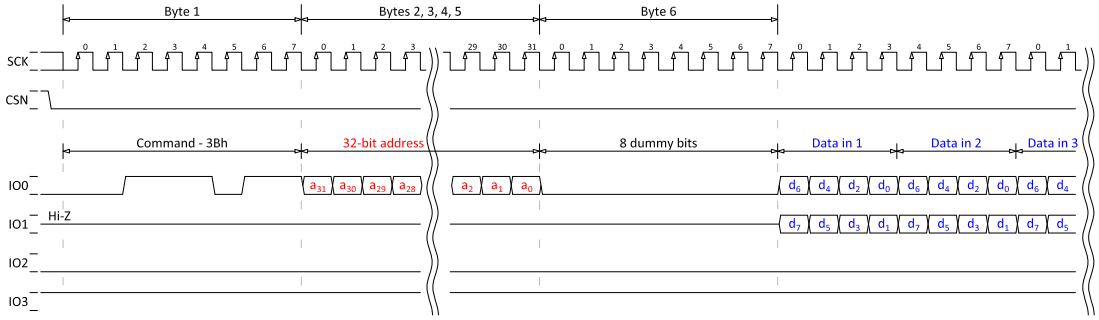

Figure 97: 32-bit READ2O (dual-read output), SPIMODE = MODE0

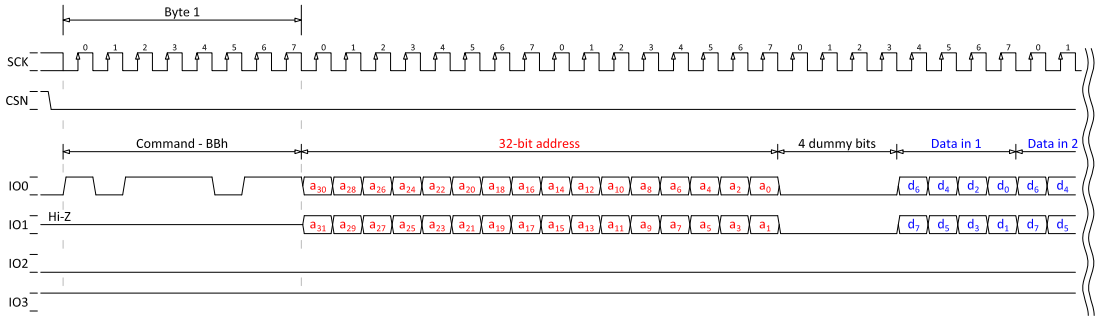

Figure 98: 32-bit READ2IO (dual read input/output), SPIMODE = MODE0

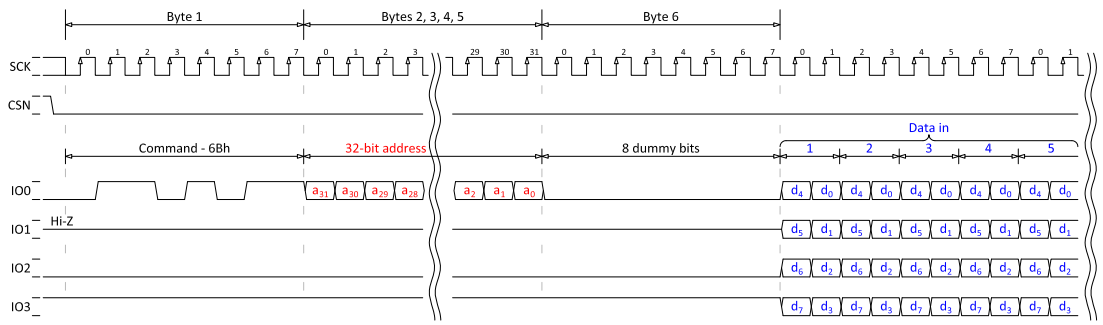

Figure 99: 32-bit READ4O (quad-read output), SPIMODE = MODE0

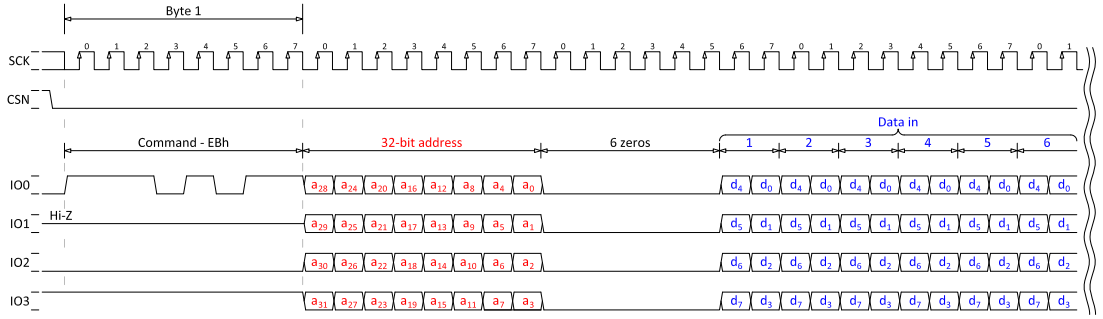

Figure 100: 32-bit READ4IO (quad-read input/output), SPIMODE = MODE0

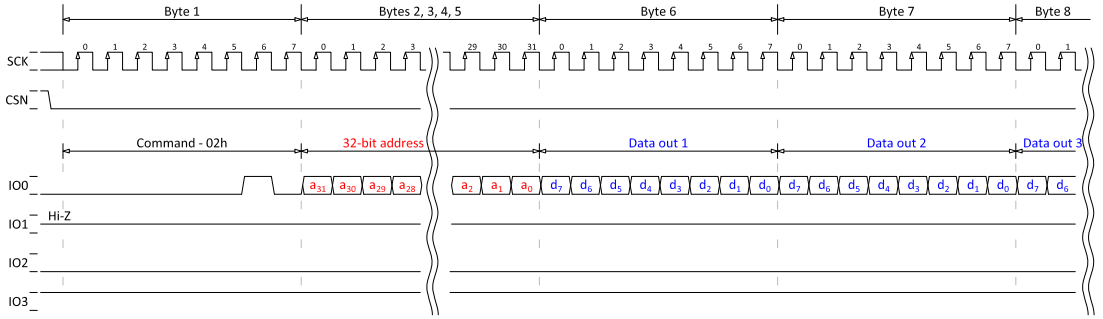

Figure 101: 32-bit PP (page program), SPIMODE = MODE0

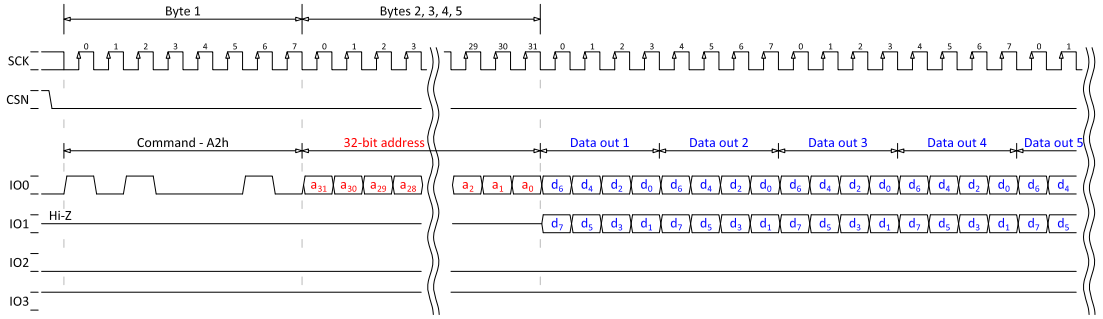

Figure 102: 32-bit PP2O (dual-page program output), SPIMODE = MODE0

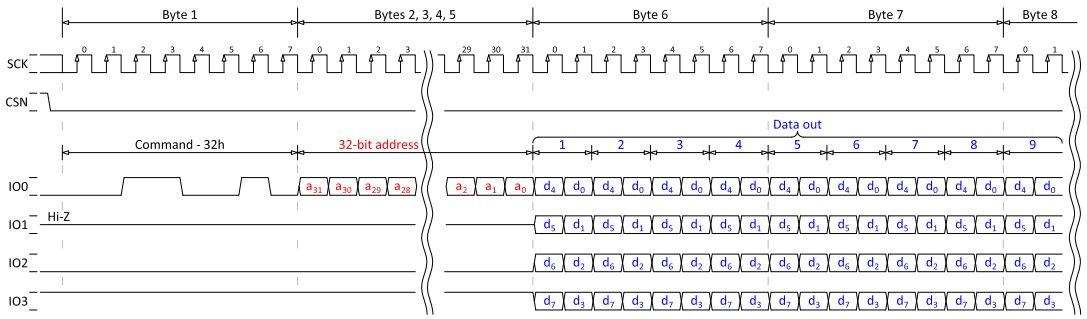

Figure 103: 32-bit PP4O (quad-page program output), SPIMODE = MODE0

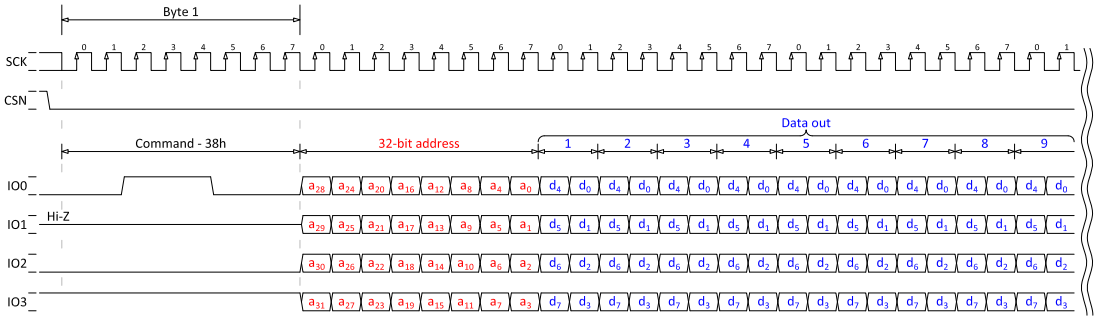

Figure 104: 32-bit PP4IO (quad page program input/output), SPIMODE = MODE0

6.19.10 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40029000	QSPI	QSPI	External memory interface	

Table 80: Instances

Register	Offset	Description
TASKS_ACTIVATE	0x000	Activate QSPI interface
TASKS_READSTART	0x004	Start transfer from external flash memory to internal RAM
TASKS_WRITESTART	0x008	Start transfer from internal RAM to external flash memory
TASKS_ERASESTART	0x00C	Start external flash memory erase operation
TASKS_DEACTIVATE	0x010	Deactivate QSPI interface
EVENTS_READY	0x100	QSPI peripheral is ready. This event will be generated as a response to any QSPI task.
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable QSPI peripheral and acquire the pins selected in PSELn registers
READ.SRC	0x504	Flash memory source address
READ.DST	0x508	RAM destination address
READ.CNT	0x50C	Read transfer length
WRITE.DST	0x510	Flash destination address
WRITE.SRC	0x514	RAM source address
WRITE.CNT	0x518	Write transfer length
ERASE.PTR	0x51C	Start address of flash block to be erased
ERASE.LEN	0x520	Size of block to be erased.
PSEL.SCK	0x524	Pin select for serial clock SCK
PSEL.CSN	0x528	Pin select for chip select signal CSN.

Register	Offset	Description
PSEL.IO0	0x530	Pin select for serial data MOSI/IO0.
PSEL.IO1	0x534	Pin select for serial data MISO/IO1.
PSEL.IO2	0x538	Pin select for serial data IO2.
PSEL.IO3	0x53C	Pin select for serial data IO3.
XIPOFFSET	0x540	Address offset into the external memory for Execute in Place operation.
IFCONFIG0	0x544	Interface configuration.
IFCONFIG1	0x600	Interface configuration.
STATUS	0x604	Status register.
DPMDUR	0x614	Set the duration required to enter/exit deep power-down mode (DPM).
ADDRCONF	0x624	Extended address configuration.
CINSTRCONF	0x634	Custom instruction configuration register.
CINSTRDAT0	0x638	Custom instruction data register 0.
CINSTRDAT1	0x63C	Custom instruction data register 1.
IFTIMING	0x640	SPI interface timing.

Table 81: Register overview

6.19.10.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Enable or disable interrupt for READY event
					See EVENTS_READY
			Disabled	0	Disable
			Enabled	1	Enable

6.19.10.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event
					See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.10.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.10.4 ENABLE

Address offset: 0x500

Enable QSPI peripheral and acquire the pins selected in PSELn registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable QSPI
			Disabled	0	Disable QSPI
			Enabled	1	Enable QSPI

6.19.10.5 READ.SRC

Address offset: 0x504

Flash memory source address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SRC			Word-aligned flash memory source address.

6.19.10.6 READ.DST

Address offset: 0x508

RAM destination address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DST			Word-aligned RAM destination address.

6.19.10.7 READ.CNT

Address offset: 0x50C

Read transfer length

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CNT		[1..0x3FFFF]	Read transfer length in number of bytes. The length must be a multiple of 4 bytes.

6.19.10.8 WRITE.DST

Address offset: 0x510

Flash destination address

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DST			Word-aligned flash destination address.

6.19.10.9 WRITE.SRC

Address offset: 0x514

RAM source address

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	SRC			Word-aligned RAM source address.

6.19.10.10 WRITE.CNT

Address offset: 0x518

Write transfer length

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CNT		[1..0x3FFFF]	Write transfer length in number of bytes. The length must be a multiple of 4 bytes.

6.19.10.11 ERASE.PTR

Address offset: 0x51C

Start address of flash block to be erased

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Word-aligned start address of block to be erased.

6.19.10.12 ERASE.LEN

Address offset: 0x520

Size of block to be erased.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LEN			LEN
			4KB	0	Erase 4 kB block (flash command 0x20)
			64KB	1	Erase 64 kB block (flash command 0xD8)
			All	2	Erase all (flash command 0xC7)

6.19.10.13 PSEL.SCK

Address offset: 0x524

Pin select for serial clock SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.14 PSEL.CSN

Address offset: 0x528

Pin select for chip select signal CSN.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.15 PSEL.IO0

Address offset: 0x530

Pin select for serial data MOSI/IO0.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.16 PSEL.IO1

Address offset: 0x534

Pin select for serial data MISO/IO1.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.17 PSEL.IO2

Address offset: 0x538

Pin select for serial data IO2.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.18 PSEL.IO3

Address offset: 0x53C

Pin select for serial data IO3.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.10.19 XIPOFFSET

Address offset: 0x540

Address offset into the external memory for Execute in Place operation.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	XIPOFFSET			Address offset into the external memory for Execute in Place operation. Value must be a multiple of 4.

6.19.10.20 IFCONFIG0

Address offset: 0x544

Interface configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G D C B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READOC			Configure number of data lines and opcode used for reading.
			FASTREAD	0	Single data line SPI. FAST_READ (opcode 0x0B).
			READ2O	1	Dual data line SPI. READ2O (opcode 0x3B).
			READ2IO	2	Dual data line SPI. READ2IO (opcode 0xBB).
			READ4O	3	Quad data line SPI. READ4O (opcode 0x6B).
			READ4IO	4	Quad data line SPI. READ4IO (opcode 0xEB).
B	RW	WRITEOC			Configure number of data lines and opcode used for writing.
			PP	0	Single data line SPI. PP (opcode 0x02).
			PP2O	1	Dual data line SPI. PP2O (opcode 0xA2).
			PP4O	2	Quad data line SPI. PP4O (opcode 0x32).
			PP4IO	3	Quad data line SPI. PP4IO (opcode 0x38).
C	RW	ADDRMODE			Addressing mode.
			24BIT	0	24-bit addressing.
			32BIT	1	32-bit addressing.
D	RW	DPMENABLE			Enable deep power-down mode (DPM) feature.
			Disable	0	Disable DPM feature.
			Enable	1	Enable DPM feature.
G	RW	PPSIZE			Page size for commands PP, PP2O, PP4O and PP4IO.
			256Bytes	0	256 bytes.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G D C B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			512Bytes	1	512 bytes.

6.19.10.21 IFCONFIG1

Address offset: 0x600

Interface configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G G G G E D A A A A A A A A
Reset 0x00040480				**0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SCKDELAY		[0..255]	Minimum amount of time that the CSN pin must stay high before it can go low again. Value is specified in number of 16 MHz periods (62.5 ns).
D	RW	DPMEN			Enter/exit deep power-down mode (DPM) for external flash memory.
			Exit	0	Exit DPM.
			Enter	1	Enter DPM.
E	RW	SPIMODE			Select SPI mode.
			MODE0	0	Mode 0: Data are captured on the clock rising edge and data is output on a falling edge. Base level of clock is 0 (CPOL=0, CPHA=0).
			MODE3	1	Mode 3: Data are captured on the clock falling edge and data is output on a rising edge. Base level of clock is 1 (CPOL=1, CPHA=1).
G	RW	SCKFREQ		[0..15]	SCK frequency is given as 32 MHz / (SCKFREQ + 1).

6.19.10.22 STATUS

Address offset: 0x604

Status register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F F F F F F F F D C
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	R	DPM			Deep power-down mode (DPM) status of external flash.
			Disabled	0	External flash is not in DPM.
			Enabled	1	External flash is in DPM.
D	R	READY			Ready status.
			READY	1	QSPI peripheral is ready. It is allowed to trigger new tasks, writing custom instructions or enter/exit DPM.
			BUSY	0	QSPI peripheral is busy. It is not allowed to trigger any new tasks, writing custom instructions or enter/exit DPM.
F	R	SREG			Value of external flash device Status Register. When the external flash has two bytes status register this field includes the value of the low byte.

6.19.10.23 DPMDUR

Address offset: 0x614

Set the duration required to enter/exit deep power-down mode (DPM).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B B B B B B B B B A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	ENTER		[0..0xFFFF]	Duration needed by external flash to enter DPM. Duration is given as ENTER * 256 * 62.5 ns.
B	RW	EXIT		[0..0xFFFF]	Duration needed by external flash to exit DPM. Duration is given as EXIT * 256 * 62.5 ns.

6.19.10.24 ADDRCONF

Address offset: 0x624

Extended address configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x000000B7				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	OPCODE		[0xFF..0]	Opcode that enters the 32-bit addressing mode.
B	RW	BYTE0		[0xFF..0]	Byte 0 following opcode.
C	RW	BYTE1		[0xFF..0]	Byte 1 following byte 0.
D	RW	MODE			Extended addressing mode.
			NoInstr	0	Do not send any instruction.
			Opcode	1	Send opcode.
			OpByte0	2	Send opcode, byte0.
			All	3	Send opcode, byte0, byte1.
E	RW	WIPWAIT			Wait for write complete before sending command.
			Disable	0	No wait.
			Enable	1	Wait.
F	RW	WREN			Send WREN (write enable opcode 0x06) before instruction.
			Disable	0	Do not send WREN.
			Enable	1	Send WREN.

6.19.10.25 CINSTRCONF

Address offset: 0x634

Custom instruction configuration register.

A new custom instruction is sent every time this register is written. The READY event will be generated when the custom instruction has been sent.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B B B B A A A A A A A A
Reset 0x00002000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OPCODE		[0..255]	Opcode of Custom instruction.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B B B B A A A A A A A A
Reset 0x00002000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
B	RW	LENGTH			Length of custom instruction in number of bytes.
			1B	1	Send opcode only.
			2B	2	Send opcode, CINSTRDAT0.BYTE0.
			3B	3	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE1.
			4B	4	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE2.
			5B	5	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE3.
			6B	6	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE4.
			7B	7	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE5.
			8B	8	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE6.
			9B	9	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE7.
C	RW	LIO2		[0..1]	Level of the IO2 pin (if connected) during transmission of custom instruction.
D	RW	LIO3		[0..1]	Level of the IO3 pin (if connected) during transmission of custom instruction.
E	RW	WIPWAIT			Wait for write complete before sending command.
			Disable	0	No wait.
			Enable	1	Wait.
F	RW	WREN			Send WREN (write enable opcode 0x06) before instruction.
			Disable	0	Do not send WREN.
			Enable	1	Send WREN.
G	RW	LFEN			Enable long frame mode. When enabled, a custom instruction transaction has to be ended by writing the LFSTOP field.
			Disable	0	Long frame mode disabled
			Enable	1	Long frame mode enabled
H	RW	LFSTOP			Stop (finalize) long frame transaction
			Stop	1	Stop

6.19.10.26 CINSTRDAT0

Address offset: 0x638

Custom instruction data register 0.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	BYTE0		[0..0xFF]	Data byte 0
B	RW	BYTE1		[0..0xFF]	Data byte 1
C	RW	BYTE2		[0..0xFF]	Data byte 2
D	RW	BYTE3		[0..0xFF]	Data byte 3

6.19.10.27 CINSTRDAT1

Address offset: 0x63C

Custom instruction data register 1.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	BYTE4		[0..0xFF]	Data byte 4
B	RW	BYTE5		[0..0xFF]	Data byte 5
C	RW	BYTE6		[0..0xFF]	Data byte 6
D	RW	BYTE7		[0..0xFF]	Data byte 7

6.19.10.28 IFTIMING

Address offset: 0x640

SPI interface timing.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C C
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	RXDELAY		[7..0]	Timing related to sampling of the input serial data. The value of RXDELAY specifies the number of 64 MHz cycles (15.625 ns) delay from the the rising edge of the SPI Clock (SCK) until the input serial data is sampled. As en example, if set to 0 the input serial data is sampled on the rising edge of SCK.

6.19.11 Electrical specification

6.19.11.1 Timing specification

Symbol	Description	Min.	Typ.	Max.	Units
$F_{QSPI,CLK}$	SCK frequency			32	MHz
$DC_{QSPI,CLK}$	SCK duty cycle				%
$F_{QSPI,XIP,16}$	XIP fetch frequency for 16 bit instructions			8	MHz
$F_{QSPI,XIP,32}$	XIP fetch frequency for 32 bit instructions			4	MHz

6.20 RADIO — 2.4 GHz radio

The 2.4 GHz radio transceiver is compatible with multiple radio standards such as 1 Mbps, 2 Mbps and long range *Bluetooth*® low energy. IEEE 802.15.4 250 kbps mode is fully supported as well as Nordic's proprietary 1 Mbps and 2 Mbps modes of operation.

Listed here are main features for the RADIO:

- Multidomain 2.4 GHz radio transceiver:
  - 1 Mbps, 2 Mbps and long range (125 kbps and 500 kbps mode) *Bluetooth*® low energy modes
  - 250 kbps IEEE 802.15.4 mode
  - 1 Mbps and 2 Mbps Nordic proprietary modes
- Best in class link budget and low power operation
- Efficient data interface with EasyDMA support
- Automatic address filtering and pattern matching

EasyDMA in combination with an automated packet assembler and packet disassembler, and an automated CRC generator and CRC checker, make it very easy to configure and use the RADIO. See RADIO block diagram on page 286 for details.

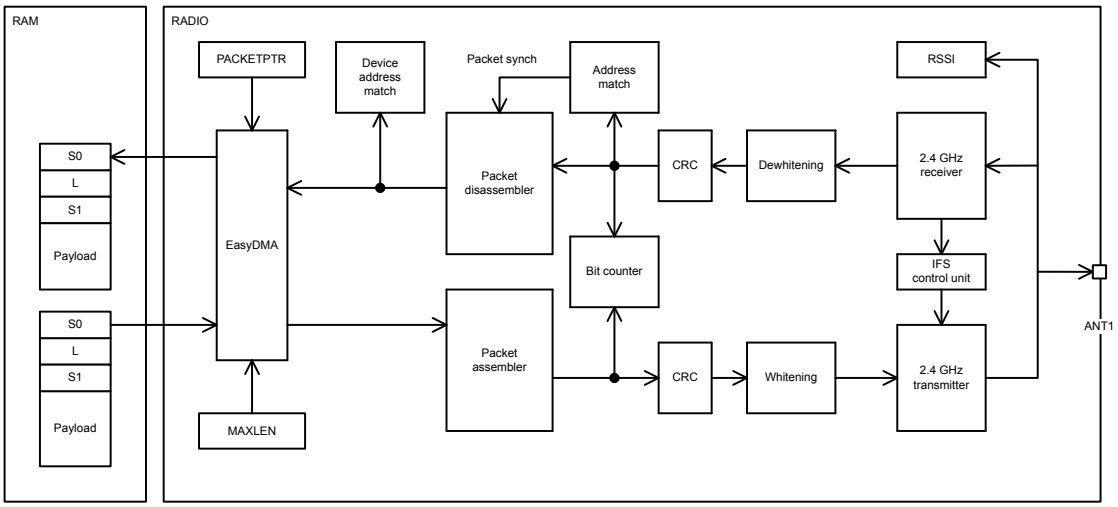

Figure 105: RADIO block diagram

The RADIO includes a device address match unit and an interframe spacing control unit that can be utilized to simplify address whitelisting and interframe spacing respectively in *Bluetooth*® low energy and similar applications.

The RADIO also includes a received signal strength indicator (RSSI) and a bit counter. The bit counter generates events when a preconfigured number of bits have been sent or received by the RADIO.

6.20.1 Packet configuration

A radio packet contains the following fields: PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD and CRC.

The content of a RADIO packet is illustrated in On air packet layout on page 286. The RADIO sends the different fields in the packet in the order they are illustrated below, from left to right:

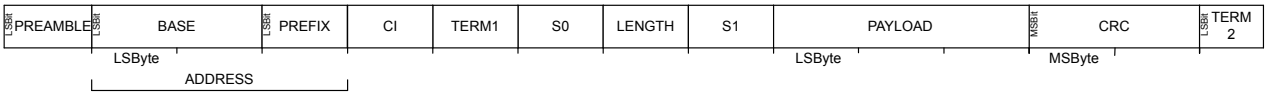

Figure 106: On air packet layout

Not shown in the figure is the static payload add-on (the length of which is defined in STATLEN, and which is 0 bytes long in a standard BLE packet). The static payload add-on is sent between PAYLOAD and CRC fields. The radio sends the different fields in the packet in the order they are illustrated above, from left to right. The preamble will be sent with least significant bit first on air.

Not shown in the figure above is the static payload add-on (the length of which is defined in PCNF1.STATLEN, and which is 0 bytes long in a standard BLE packet). The static payload add-on is sent between the PAYLOAD and CRC fields.

PREAMBLE is sent with least significant bit first on-air. The size of the PREAMBLE depends on the mode selected in the MODE register:

- For all Nordic proprietary radio modes (Nrf_1Mbit, Nrf_2Mbit and Nrf_250Kbit) and for the Ble_1Mbit mode, the PREAMBLE is one byte long. The PLEN field in the PCNF0 register has to be set accordingly. If the first bit of the ADDRESS is 0, the PREAMBLE is set to 0xAA. Otherwise the PREAMBLE is set to 0x55.

- For the Ble_2Mbit mode, the PREAMBLE is 2 bytes long. The PLEN field in the PCNF0 register has to be set to 2 bytes accordingly. If the first bit of the ADDRESS is 0, the PREAMBLE is set to 0xAAAA. Otherwise the PREAMBLE is set to 0x5555.
- For the modes Ble_LR125Kbit and Ble_LR500Kbit, the PREAMBLE is 10 repetitions of 0x3C.
- For the Ieee802154_250Kbit mode, the PREAMBLE is 4 bytes long and set to all zeros.

The PREAMBLE is one byte long for Nordic proprietary operating modes, and the PLEN field in the PCNF0 register has to be set accordingly. If the first bit of the ADDRESS is 0 the preamble will be set to 0xAA otherwise the PREAMBLE will be set to 0x55.

For MODE = Ble_2Mbit the PREAMBLE has to be set to 2 byte long through the PLEN field in the PCNF0 register. If the first bit of the ADDRESS is 0 the preamble will be set to 0xAAAA otherwise the PREAMBLE will be set to 0x5555.

For MODE = Ble_LR125Kbit and MODE = Ble_LR500Kbit the PREAMBLE is 10 repetitions of 0x3C.

For MODE = Ieee802154_250Kbit the PREAMBLE is 4 bytes long and set to all zeros.

Radio packets are stored in memory inside instances of a radio packet data structure as illustrated in In-RAM representation of radio packet - S0, LENGTH and S1 are optional on page 287. The PREAMBLE, ADDRESS, CI, TERM1, TERM2 and CRC fields are omitted in this data structure.

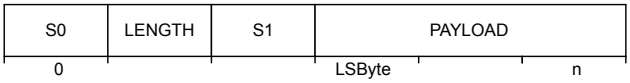

Figure 107: In-RAM representation of radio packet - S0, LENGTH and S1 are optional

The byte ordering on air is always least significant byte first for the ADDRESS and PAYLOAD fields and most significant byte first for the CRC field. The ADDRESS fields are always transmitted and received least significant bit first on air. The CRC field is always transmitted and received most significant bit first. The bit-endian, i.e. the order in which the bits are sent and received, of the S0, LENGTH, S1 and PAYLOAD fields can be configured via the ENDIAN in PCNF1.

The sizes of the S0, LENGTH and S1 fields can be individually configured via S0LEN, LFLEN and S1LEN in PCNF0 respectively. If any of these fields are configured to be less than 8 bits long, the least significant bits of the fields are used.

If S0, LENGTH or S1 are specified with zero length their fields will be omitted in memory, otherwise each field will be represented as a separate byte, regardless of the number of bits in their on air counterpart.

Independent of the configuration of MAXLEN, the combined length of S0, LENGTH, S1 and PAYLOAD cannot exceed 258 bytes.

6.20.2 Address configuration

The on air radio ADDRESS field is composed of two parts, the base address field and the address prefix field.

The size of the base address field is configurable via BALEN in PCNF1. The base address is truncated from the least significant byte if the BALEN is less than 4. See Definition of logical addresses on page 288.

Logical address	Base address	Prefix byte
0	BASE0	PREFIX0.AP0
1	BASE1	PREFIX0.AP1
2	BASE1	PREFIX0.AP2
3	BASE1	PREFIX0.AP3
4	BASE1	PREFIX1.AP4
5	BASE1	PREFIX1.AP5
6	BASE1	PREFIX1.AP6
7	BASE1	PREFIX1.AP7

Table 82: Definition of logical addresses

The on air addresses are defined in the BASEn and PREFIXn registers, and it is only when writing these registers the user will have to relate to actual on air addresses. For other radio address registers such as the TXADDRESS, RXADDRESSES and RXMATCH registers, logical radio addresses ranging from 0 to 7 are being used. The relationship between the on air radio addresses and the logical addresses is described in Definition of logical addresses on page 288.

6.20.3 Data whitening

The RADIO is able to do packet whitening and de-whitening.

See WHITEEN in PCNF1 register for how to enable whitening. When enabled, whitening and de-whitening will be handled by the RADIO automatically as packets are sent and received.

The whitening word is generated using polynomial $g(D) = D^7 + D^4 + 1$, which then is XORed with the data packet that is to be whitened, or de-whitened. See the figure below.

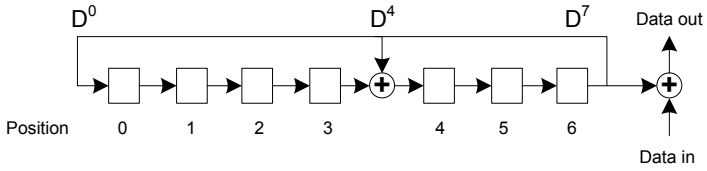

Figure 108: Data whitening and de-whitening

Whitening and de-whitening will be performed over the whole packet (except for the preamble and the address field).

The linear feedback shift register, illustrated in Data whitening and de-whitening on page 288 can be initialised via the DATAWHITEIV register.

6.20.4 CRC

The CRC generator in the RADIO calculates the CRC over the whole packet excluding the preamble. If desirable, the address field can be excluded from the CRC calculation as well

See CRCCNF register for more information.

The CRC polynomial is configurable as illustrated in CRC generation of an n bit CRC on page 289 where bit 0 in the CRCPOLY register corresponds to X^0 and bit 1 corresponds to X^1 etc. See CRCPOLY for more information.

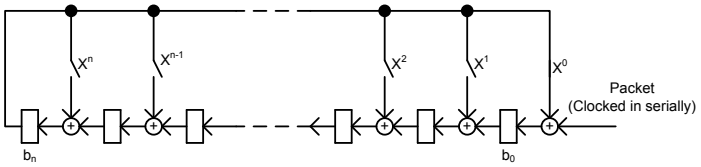

Figure 109: CRC generation of an n bit CRC

As illustrated in CRC generation of an n bit CRC on page 289, the CRC is calculated by feeding the packet serially through the CRC generator. Before the packet is clocked through the CRC generator, the CRC generator's latches b_0 through b_n will be initialized with a predefined value specified in the CRCINIT register. When the whole packet is clocked through the CRC generator, latches b_0 through b_n will hold the resulting CRC. This value will be used by the RADIO during both transmission and reception but it is not available to be read by the CPU at any time. A received CRC can however be read by the CPU via the RXCRC register independent of whether or not it has passed the CRC check.

The length (n) of the CRC is configurable, see CRCCNF for more information.

After the whole packet including the CRC has been received, the RADIO will generate a CRCOK event if no CRC errors were detected, or alternatively generate a CRCERROR event if CRC errors were detected.

The status of the CRC check can be read from the CRCSTATUS register after a packet has been received.

6.20.5 Radio states

Tasks and events are used to control the operating state of the RADIO.

The RADIO can enter the states described the table below.

State	Description
DISABLED	No operations are going on inside the radio and the power consumption is at a minimum
RXRU	The radio is ramping up and preparing for reception
RXIDLE	The radio is ready for reception to start
RX	Reception has been started and the addresses enabled in the RXADDRESSES register are being monitored
TXRU	The radio is ramping up and preparing for transmission
TXIDLE	The radio is ready for transmission to start
TX	The radio is transmitting a packet
RXDISABLE	The radio is disabling the receiver
TXDISABLE	The radio is disabling the transmitter

Table 83: RADIO state diagram

An overview state diagram for the RADIO is illustrated in Radio states on page 290.

Note: PHYEND is only generated in Ble_LR125Kbit, Ble_LR500Kbit and Ieee802154_250Kbit modes.

Note: The END to START shortcut should not be used with Ble_LR125Kbit, Ble_LR500Kbit and Ieee802154_250Kbit modes. Rather the PHYEND to START shortcut.

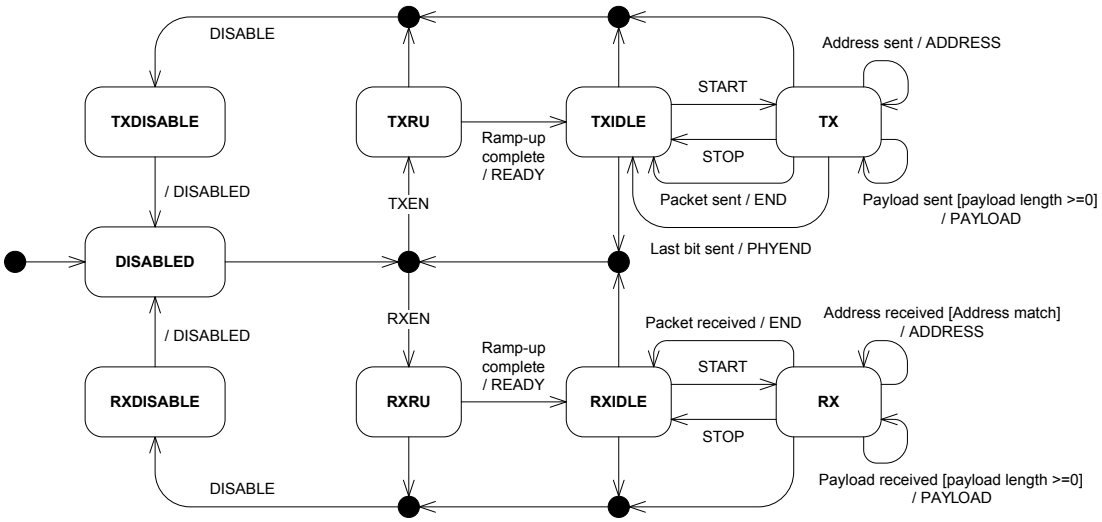

Figure 110: Radio states

This figure shows how the tasks and events relate to the RADIO's operation. The RADIO does not prevent a task from being triggered from the wrong state. If a task is triggered from the wrong state, for example if the RXEN task is triggered from the RXDISABLE state, this may lead to incorrect behaviour. As illustrated in Radio states on page 290, the PAYLOAD event is always generated even if the payload is zero.

6.20.6 Transmit sequence

Before the RADIO is able to transmit a packet, it must first ramp-up in TX mode.

See TXRU in Radio states on page 290 and Transmit sequence on page 290. A TXRU ramp-up sequence is initiated when the TXEN task is triggered. After the radio has successfully ramped up it will generate the READY event indicating that a packet transmission can be initiate. A packet transmission is initiated by triggering the START task. As illustrated in Radio states on page 290 the START task can first be triggered after the RADIO has entered into the TXIDLE state.

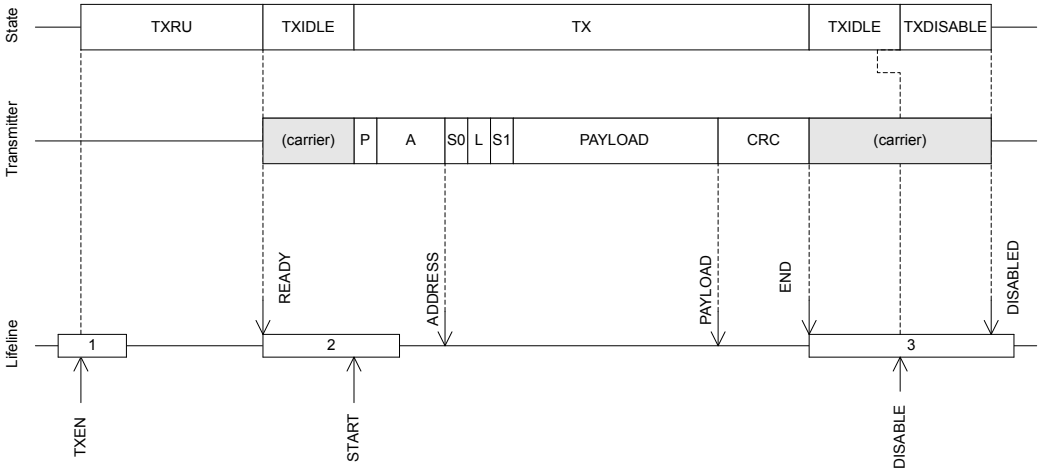

Figure 111: Transmit sequence

Transmit sequence on page 290 illustrates a single packet transmission where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay caused by CPU execution is expected between READY and START, and between END and DISABLE. As illustrated in Transmit sequence on page 290 the RADIO will by default transmit '1's between READY and START, and between END and DISABLED. What is transmitted can be programmed through the DTX field in the MODECNF0 register.

A slightly modified version of the transmit sequence from Transmit sequence on page 290 is illustrated in Transmit sequence using shortcuts to avoid delays on page 291 where the RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.

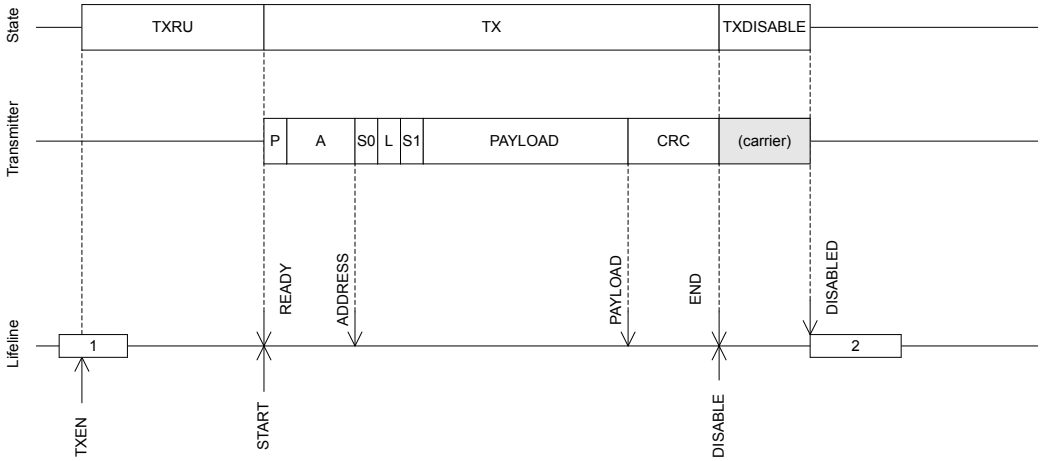

Figure 112: Transmit sequence using shortcuts to avoid delays

The RADIO is able to send multiple packets one after the other without having to disable and re-enable the RADIO between packets, this is illustrated in Transmission of multiple packets on page 291.

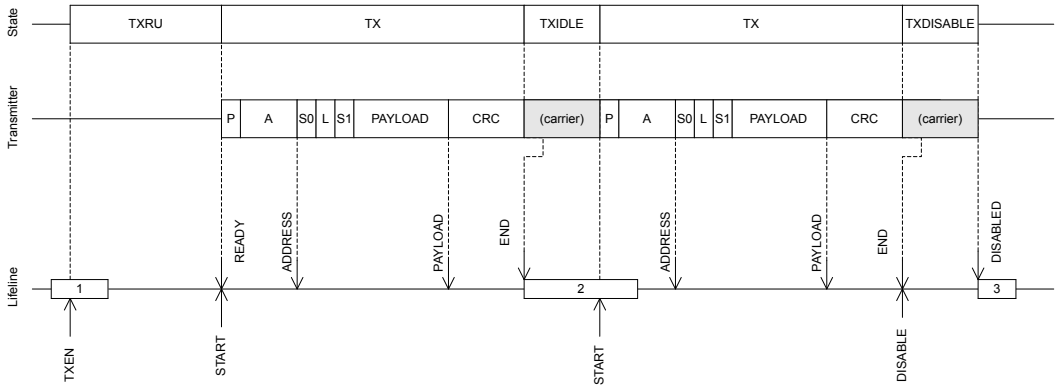

Figure 113: Transmission of multiple packets

6.20.7 Receive sequence

Before the RADIO is able to receive a packet, it must first ramp up in RX mode

See RXRU in Radio states on page 290 and Receive sequence on page 292.

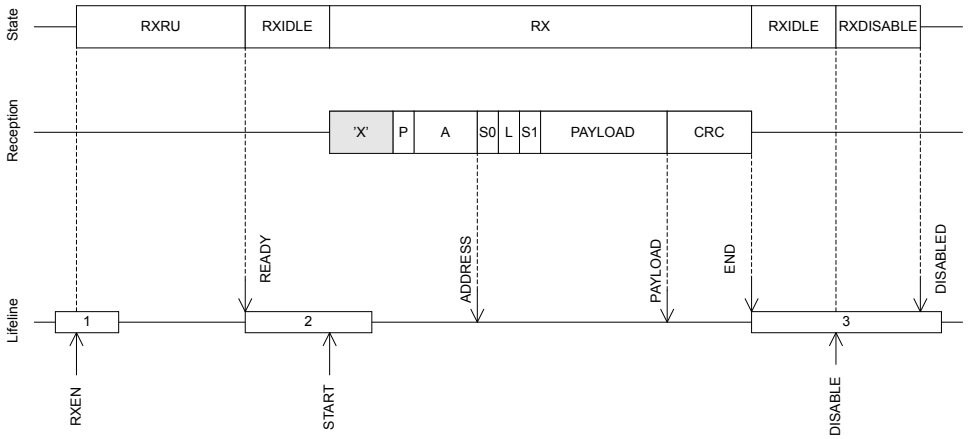

Figure 114: Receive sequence

An RXRU ramp up sequence is initiated when the RXEN task is triggered. After the radio has successfully ramped up it will generate the READY event indicating that a packet reception can be initiated. A packet reception is initiated by triggering the START task. As illustrated in Radio states on page 290 the START task can first be triggered after the RADIO has entered into the RXIDLE state.

Receive sequence on page 292 illustrates a single packet reception where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay caused by CPU execution is expected between READY and START, and between END and DISABLE. As illustrated Receive sequence on page 292 the RADIO will be listening and possibly receiving undefined data, represented with an 'X', from START and until a packet with valid preamble (P) is received.

A slightly modified version of the receive sequence from Receive sequence on page 292 is illustrated in Receive sequence using shortcuts to avoid delays on page 292 where the RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.

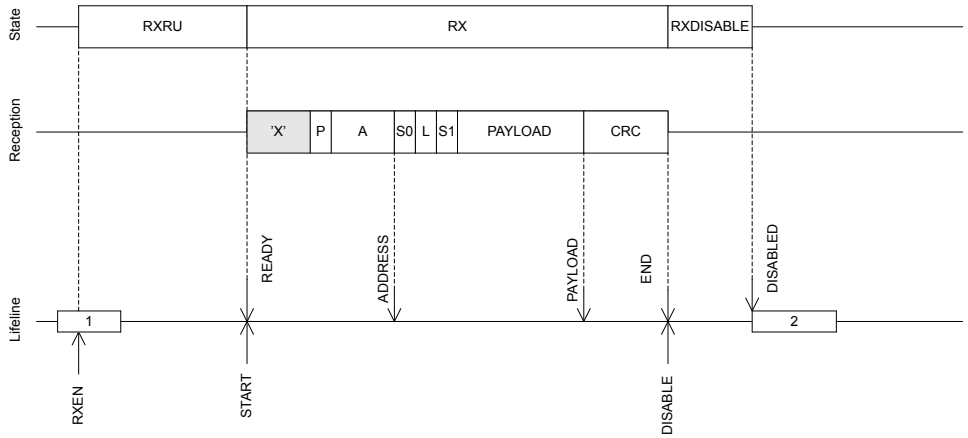

Figure 115: Receive sequence using shortcuts to avoid delays

The RADIO is able to receive multiple packets one after the other without having to disable and re-enable the RADIO between packets as illustrated in Reception of multiple packets on page 293.

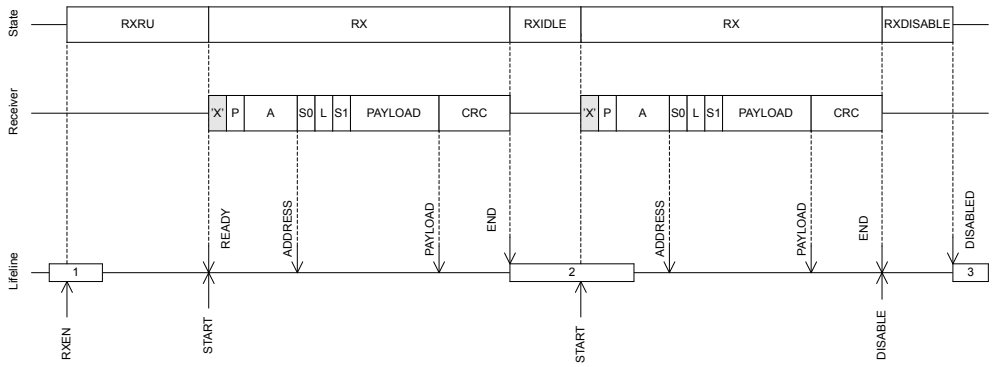

Figure 116: Reception of multiple packets

6.20.8 Received signal strength indicator (RSSI)

The radio implements a mechanism for measuring the power in the received radio signal. This feature is called received signal strength indicator (RSSI).

Sampling of the received signal strength is started by using the RSSISTART task. The sample can be read from the RSSISAMPLE register.

The sample period of the RSSI is defined by $RSSI_{PERIOD}$. The RSSI sample will hold the average received signal strength during this sample period.

For the RSSI sample to be valid the radio has to be enabled in receive mode (RXEN task) and the reception has to be started (READY event followed by START task).

6.20.9 Interframe spacing

Interframe spacing is the time interval between two consecutive packets.

It is defined as the time, in microseconds, from the end of the last bit of the previous packet received and to the start of the first bit of the subsequent packet that is transmitted. The RADIO is able to enforce this interval, as specified in the TIFS register, as long as the TIFS is not specified to be shorter than the RADIO's turnaround time, i.e. the time needed to switch off the receiver, and then switch the transmitter back on. The TIFS register can be written any time before the last bit on air is received.

This timing is illustrated in the figure below.

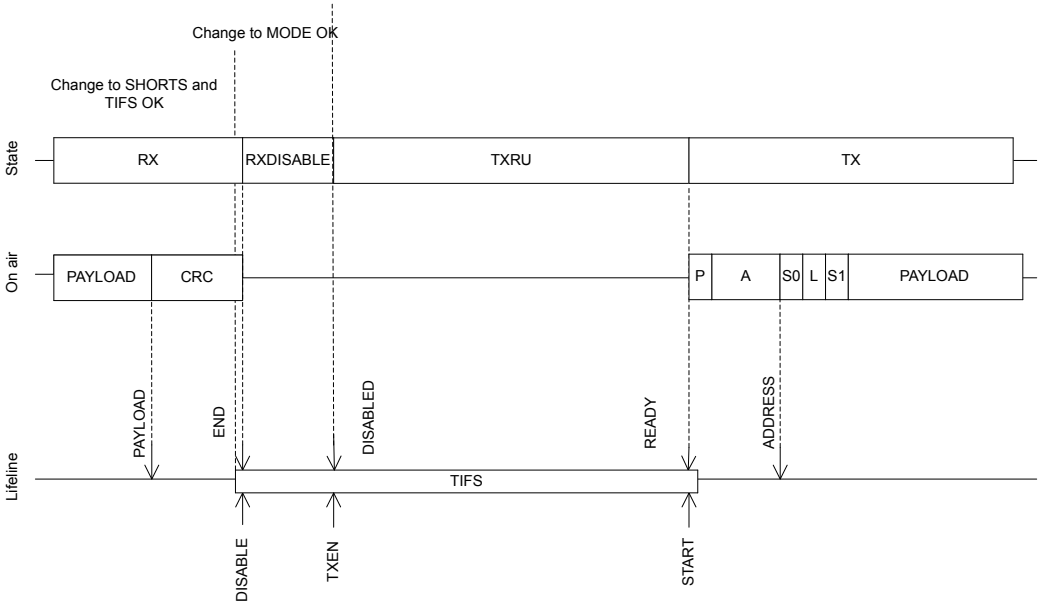

Figure 117: IFS timing detail

As illustrated, the TIFS duration starts after the last bit on air (just before the END event), and elapses with first bit being transmitted on air (just after READY event).

TIFS is only enforced if END_DISABLE and DISABLED_TXEN or END_DISABLE and DISABLED_RXEN shortcuts are enabled. TIFS is qualified for use in BLE_1MBIT, BLE_2MBIT, BLE_LR125KBIT, BLE_LR500KBIT and Ieee802154_250Kbit mode using the default ramp-up mode. SHORTS and TIFS are not double-buffered, and can be updated at any point in time before the last bit on air is received. The MODE register is double-buffered and sampled at the TXEN or RXEN task.

6.20.10 Device address match

The device address match feature is tailored for address whitelisting in a *Bluetooth*® low energy and similar implementations.

This feature enables on-the-fly device address matching while receiving a packet on air. This feature only works in receive mode and as long as RADIO is configured for little endian, see PCNF1.ENDIAN.

The device address match unit assumes that the 48 first bits of the payload is the device address and that bit number 6 in S0 is the TxAdd bit. See the *Bluetooth*® Core Specification for more information about device addresses, TxAdd and whitelisting.

The RADIO is able to listen for eight different device addresses at the same time. These addresses are specified in a DAB/DAP register pair, one pair per address, in addition to a TxAdd bit configured in the DACNF register. The DAB register specifies the 32 least significant bits of the device address, while the DAP register specifies the 16 most significant bits of the device address.

Each of the device addresses can be individually included or excluded from the matching mechanism. This is configured in the DACNF register.

6.20.11 Bit counter

The RADIO implements a simple counter that can be configured to generate an event after a specific number of bits have been transmitted or received.

By using shortcuts, this counter can be started from different events generated by the RADIO and hence count relative to these.

The bit counter is started by triggering the BCSTART task, and stopped by triggering the BCSTOP task. A BCMATCH event will be generated when the bit counter has counted the number of bits specified in the

BCC register. The bit counter will continue to count bits until the DISABLED event is generated or until the BCSTOP task is triggered. The CPU can therefore, after a BCMATCH event, reconfigure the BCC value for new BCMATCH events within the same packet.

The bit counter can only be started after the RADIO has received the ADDRESS event.

The bit counter will stop and reset on BCSTOP, STOP, END and DISABLE tasks.

The figure below illustrates how the bit counter can be used to generate a BCMATCH event in the beginning of the packet payload, and again generate a second BCMATCH event after sending 2 bytes (16 bits) of the payload.

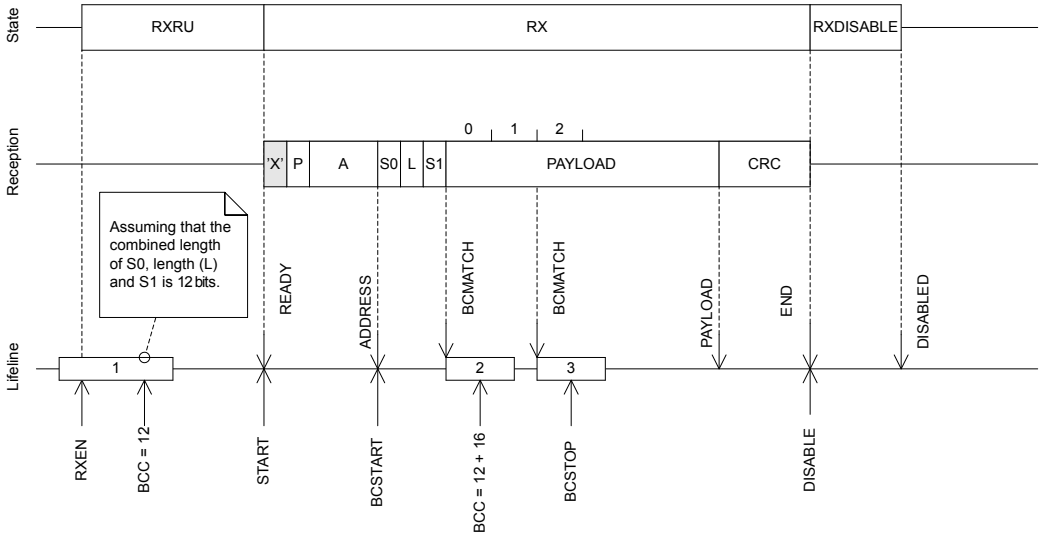

Figure 118: Bit counter example

6.20.12 IEEE 802.15.4 operation

With the MODE=Ieee802154_250kbit the radio module will comply with the IEEE 802.15.4-2006 standard implementing its 250 kbps 2450MHz O-QPSK PHY.

The IEEE 802.15.4 standard differs from Nordic's proprietary and *Bluetooth*® low energy modes. Obvious differences are modulation scheme and channel structure, but also packet structure, security and medium access control.

The main features of the IEEE 802.15.4 mode are:

- Ultra low power 250 kbps 2450MHz IEEE 802.15.4-2006 compliant link
- Clear channel assessment
- Energy detection scan
- CRC generation

6.20.12.1 Packet structure

The IEEE 802.15.4 standard defines an on the air frame/packet that is different from what is used in BLE mode.

The following figure provides an overview of the physical frame structure and its timing:

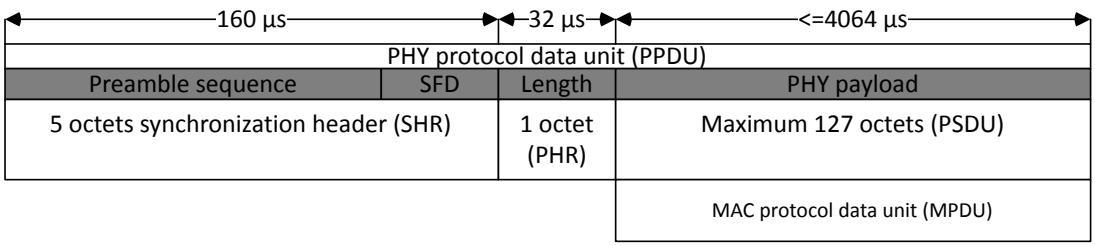

Figure 119: IEEE 802.15.4 frame format - PHY layer frame structure (PPDU)

The standard uses the term octet as storage unit for 8 bits within the PPDU. For timing, the value symbol is used, and it has the duration of 16 µs.

The total usable payload (PSDU) is 127 octets, but when CRC is being used, this is reduced to 125 octets of usable payload.

The preamble sequence consists of four octets that are all zero. These are used for the radio receiver to synchronize on. Following the four octets is a single octet named start of frame delimiter (SFD) with a fixed value of 0xA7. The user can program an alternative SFD through the SFD register. This feature is provided for an initial level of frame filtering for those who choose non-standard compliance. It is a valuable feature when operating in a congested or private network. The preamble sequence and the SFD are generated by the radio module, and are not programmed by the user into the frame buffer.

The PHY header (PHR) is a single octet following the synchronization header (SHR). The least significant seven bits denote the frame length of the following PSDU. The most significant bit is reserved and is set to zero for frames that are standard compliant. The radio module will report all eight bits and it can potentially be used to carry some information. The PHR is the first byte that will be written to the frame data memory pointed to by PACKETPTR. Frames with zero length will be discarded, and the FRAMESTART event will not be generated in this case.

The next N octets will carry the data of the PHY packet, where N equals the value of the PHR. For an implementation also using the IEEE 802.15.4 MAC layer, the PHY data will be a MAC frame of N-2 octets since two octets will occupy a CRC field.

An IEEE 802.15.4 MAC frame will always consist of a header (the frame control field (FCF), sequence number and addressing fields), a payload, and the 16-bit frame control sequence (FCS), as as illustrated in the figure below.

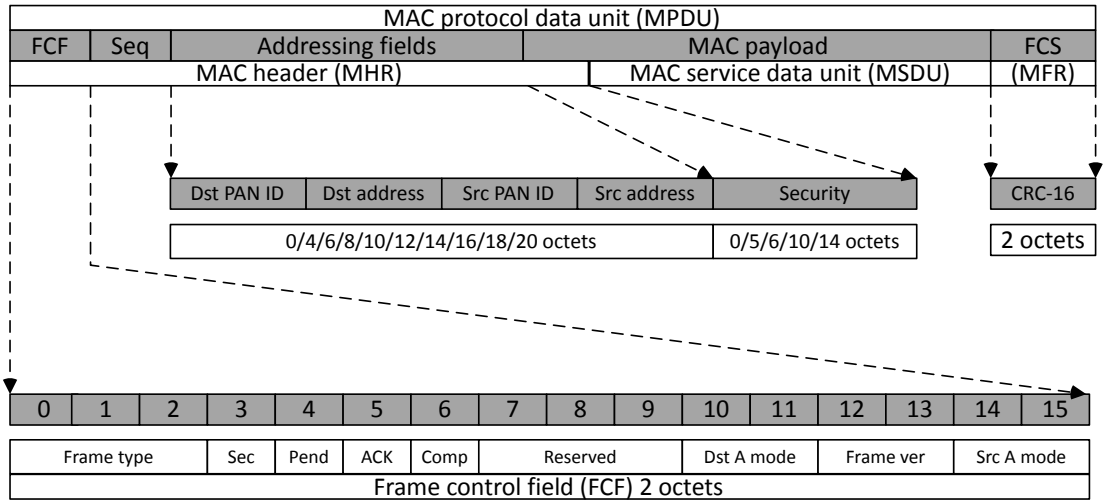

Figure 120: IEEE 802.15.4 frame format - MAC layer frame structure (MPDU)

The two FCF octets contain information about what type of frame this is, what addressing it uses, and other control flags. This field is decoded when using the assisted operating modes offered by the radio.

The sequence number is a single octet in size and is unique for a frame. It will be used in the associated acknowledgement frame sent upon successful frame reception.

The addressing field can be zero (acknowledgement frame) or up to 20 octets in size. The field is used to direct packets to the correct recipient as well as denoting its origin. IEEE 802.15.4 bases it's addressing on networks being organized in PANs with 16-bit identifier and nodes having a 16-bit or 64-bit address. In the assisted receive mode, these parameters are analyzed for address matching and acknowledgement.

The MAC payload carries the data of the next higher layer, or in the case of a MAC command frame information used by the MAC layer itself.

The two last octets contain the 16-bit ITU-T CRC. The FCS is calculated over the MAC header (MHR) and MAC payload (MSDU) parts of the frame. This field is calculated automatically when sending a frame, or indicated in the CRCSTATUS register when a frame is received. This feature is taken care of autonomously, by the CRC module (if configured).

6.20.12.2 Operating frequencies

The IEEE 802.15.4 standard defines 16 channels [11 - 26] of 5 MHz each in the 2450 MHz frequency band.

The FREQUENCY register of the radio module must be programmed according to table below for correct operation on the center frequency defined for each channel.

IEEE 802.15.4 channel	Center frequency (MHz)	FREQUENCY setting
Channel 11	2405	5
Channel 12	2410	10
Channel 13	2415	15
Channel 14	2420	20
Channel 15	2425	25
Channel 16	2430	30
Channel 17	2435	35
Channel 18	2440	40
Channel 19	2445	45
Channel 20	2450	50
Channel 21	2455	55
Channel 22	2460	60
Channel 23	2465	65
Channel 24	2470	70
Channel 25	2475	75
Channel 26	2480	80

Table 84: IEEE 802.15.4 center frequency definition

6.20.12.3 Energy detection (ED)

The IEEE 802.15.4 standard requires that it is possible to sample the received signal power within the bandwidth of a channel for the purpose of determining presence of activity.

There should be no attempt made to decode the signals on the channel, and this is done by disabling the shortcut between READY event and START task before putting the radio in receive mode. The energy detection (ED) measurement time where RSSI samples are averaged over is 8 symbol periods (128 µs). The standard further specifies the measurement to be a number between 0 and 0xFF - where 0 shall indicate received power less than 10 dB above the selected receiver sensitivity. The power range of the ED values must be at least 40 dB with a linear mapping with accuracy of ±6 dB. See section *6.9.7 Receiver ED* in the IEEE 802.15.4 standard for further details. An example of an ED scan is given below.

Below is a code snippet showing how to perform a single energy detection measurement.

```
uint8_t sample_ed(void)
{
 NRF_RADIO->TASKS_EDSTART = 1;              //Start
 while (NRF_RADIO->EVENTS_EDEND != 1) {
     //CPU can sleep here or do something else
     //Use of interrupts are encouraged
   }
 return (uint8_t)NRF_RADIO->EDSAMPLE; //Read level
}
```

It is the mlme-scan.req primitive of the MAC layer that is using the ED measurement to detect channels where there might be wireless activity. To assist this primitive a taylored mode of operation is available where the ED measurement runs for a defined number of iterations where it keeps track of the maximum ED level. This is enganged by writing the ED_CNT register to a value different from 0, it will then run the specified number of iterations reporting the maximum energy measurement in the EDSAMPLE register. The scan is started with EDSTART task and its end indicated with the EDEND event. This greatly reduces the interrupt frequency and hence power consumtion. The figure below shows how the ED measurement will operate depending on the ED_CNT register.

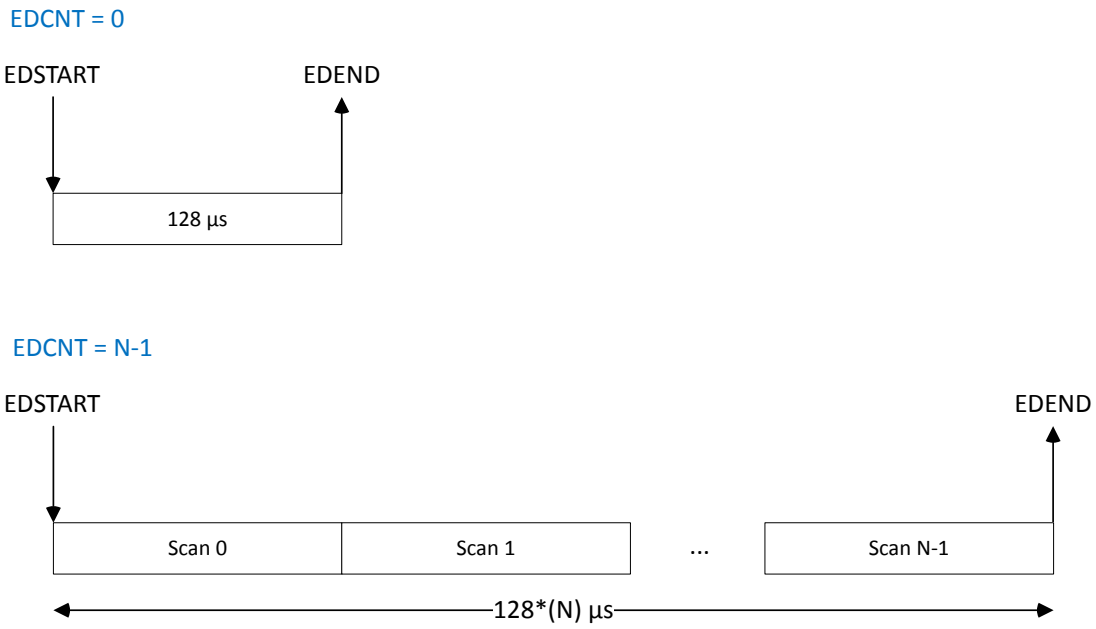

Figure 121: Energy detection measurement examples

An ongoing scan can always be stopped by writing the EDSTOP task. It will be followed by the EDSTOPPED event when the module has terminated.

6.20.12.4 Clear channel assessment (CCA)

IEEE 802.15.4 implements a listen-before-talk channel access method to avoid collisions when transmitting - namely carrier sense multiple access with collision avoidance (CSMA-CA). The key part of this is measuring if the wireless medium is busy or not.

At least three methods must be supported:

- Mode 1 (energy above threshold): The medium is reported busy upon detecting any energy above the ED threshold
- Mode 2 (carrier sense only): The medium is reported busy upon detection of a signal compliant with the IEEE 802.15.4 standard with the same modulation and spreading characteristics
- Mode 3 (carrier sense and threshold): The medium is reported busy by logically ANDing or ORing the results from mode 1 and mode 2.

It is furthermore specified that the clear channel assessment should survey a period equal to 8 symbols or 128 µs.

The radio module has to be in receive mode and be able to recived correct packets when performing the CCA. The shortcut between READY and START must be disabled if baseband processing is not to be performed while the measurement is running.

Mode 1 is enabled by first configuring the CCA_MODE=EdMode and writing the CCA_EDTHRES to a chosen value. When the CCA_START task is written the radio module will perform a ED measurement for 8 symbols and compare the measured level with that found in the CCA_EDTHRES register. If the measured value is higher than or equal to this threshold the CCABUSY event is generated - the CCAIDLE event is generated if the measured level is less than the threshold.

Mode 2 is enabled by configuring the CCA_MODE=CarrierMode. In carrier mode the module will sample to see if a valid SFD is found during the 8 symbols. If a valid SFD is seen the CCABUSY event is generated and the node should not send any data. The CCABUSY event is also generated if the scan was performed during an ongoing frame reception. In the case where the measurement period completes with no SFD detection the CCAIDLE task is generated. With the CCA_CORR_COUNT unequal to zero the algorithm will look at the correlator output in addition to the SFD detection signal. If a SFD is reported during the scan period it will terminate immidiately indicating busy medium. Similarly, if the number of peaks above CCA_CORRTHRES crosses the CCA_CORR_COUNT the CCABUSY event is generated. If less than CCA_CORR_COUNT crossings are found and no SFD is reported the CCAIDLE signal will be generated and it is ok for the node to commence sending data.

With the CCA_MODE=CarrierAndEdMode or CCA_MODE=CarrierOrEdMode a logical combination of the result from running both mode 1 and mode 2 is performed. The CCABUSY or CCAIDLE signal will be generated based on an ANDing or ORing of the internal signals from performing both the energy detection and carrier detection scans.

An ongoing CCA can always be stopped by issuing the CCASTOP task. This will trigger the associated CCASTOPPED event.

For CCA mode automation there are three shortcuts available. One is between CCAIDLE and TXEN. This short must always be used in conjunction with the short between CCAIDLE and STOP. This automation is provided so that the radio can automatically switch between RX (when performing the CCA) and to TX where the packet is sent. The last shortcut associated with the CCA mode is between CCABUSY and DISABLE. This will cause the radio to be disabled whenever the CCA reports a busy medium.

Another handy shortcut is between RXREADY and CCASTART. When the radio has ramped up into RX mode it can immidiately start a CCA.

6.20.12.5 Cyclic redundancy check (CRC)

IEEE 802.15.4 uses a 16-bit ITU-T cyclic redundancy check (CRC) calculated over the MAC header (MHR) and MAC service data unit (MSDU).

The standard defines the following generator polynomial:

$G(x) = x^{16} + x^{12} + x^5 + 1$

In receive mode the radio will trigger the CRC module when the first octet after the frame length (PHR) is received. The CRC will then update on each consecutive octet received. When a complete frame is received the CRCSTATUS register will be updated accordingly and the EVENTS_CRCOK or EVENTS_CRCERROR generated. When the CRC module is enabled it will not write the two last octets (CRC)

to the frame Data RAM. When transmitting the CRC will be computed on the fly, starting with the first octet after PHR, and inserted as the two last octets in the frame. The EasyDMA will fetch frame length - 2 octets from DataRAM and insert the CRC octets insitu.

Below is a code snippet for configuring the CRC module for correct operation when in IEEE 802.15.4 mode. The CRCCNF is written to 16-bit CRC and the CRCPOLY is written to 0x121. The start value used by IEEE 802.15.4 is zero and CRCINIT is configured to reflect this.

```
/* 16-bit CRC with ITU-T polynomial with 0 as start condition*/
write_reg(NRFRADIO_REG(CRCCNF), 0x202);
write_reg(NRFRADIO_REG(CRCPOLY), 0x11021);
write_reg(NRFRADIO_REG(CRCINIT), 0);
```

The ENDIANESS subregister must be set to little-endian since the FCS field is transmitted leftmost bit first.

6.20.12.6 Transmit sequence

The transmission is started by first putting the radio in receive mode sending the RXEN task.

An outline of the IEEE 802.15.4 transmission is illustrated in the figure below.

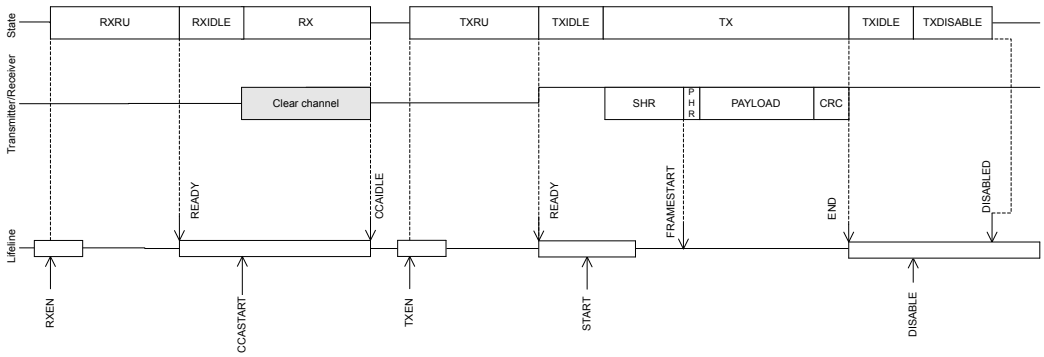

Figure 122: IEEE 802.15.4 transmit sequence

The receiver will ramp up and enter the RXIDLE state where the READY event is generated. Upon receiving the ready event the CCA is started by writing to the CCASTART task register. The chosen mode of assessment (CCA_MODE register) will be performed and signal the CCAIDLE or CCABUSY event 128 µs later. If the CCABUSY is received the radio will have to retry the CCA after a specific back off period as outlined in the IEEE 802.15.4 standard (see Figure 69 in section *7.5.1.4 The CSMA-CA algorithm* of the standard).

When the CCAIDLE event on the other hand is generated the user shall write to the TXEN task register to enter the TXRU state. The READY event will be generated when the radio is in TXIDLE state and ready to transmit. With the PACKETPTR pointing to the length (PHR) field of the frame the START task can be written. The radio will send the four octet preamble sequence followed by the start of frame delimiter (SFD register). The first byte read from the Data RAM is the length field (PHR) followed by the transmission of the number of bytes indicated as the frame length. If the CRC module is configured it will run for PHR-2 octets. The last two octets will be substituted with the results from running the CRC. The necessary CRC parameters are sampled on the START task. The FCS field of the frame is little endian.

In addition to the already available shortcuts, one is provided between READY event and CCASTART task so that a CCA can automatically start when the receiver is ready. And a second shortcut has been added between CCAIDLE event and the TXEN task so that upon detecting a clear channel the radio can immediately enter transmit mode.

6.20.12.7 Receive sequence

The reception is started by first putting the radio in receive mode. Writing to the RXEN task the radio will start ramping up and enter the RXRU state.

When the READY event is generated the radio has entered the RXIDLE mode. For the baseband processing to be enabled the START task must be written. An outline of the IEEE 802.15.4 reception can be found in figure below.

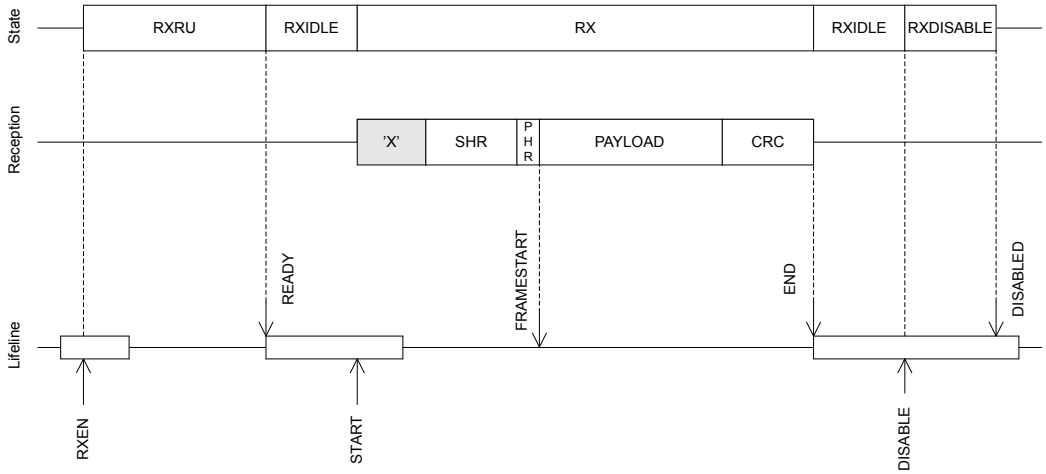

Figure 123: IEEE 802.15.4 receive sequence

When a valid SHR is received the radio will start storing future octets (starting with PHR) to the data memory pointed to by PACKETPTR. After the SFD octet is received the FRAMESTART event is generated. If the CRC module is enabled it will start updating with the second byte received (first byte in payload) and run for the full frame length. The two last bytes in the frame is not written to DataRAM when CRC is configured. However, if the result of the CRC after running the full frame is zero the CRCOK event will be generated. The END event is generated when the last octet has been received and is available in DataRAM.

When a packet is received a link quality indicator (LQI) is also generated and appended immediately after the last received octet. When using IEEE 802.15.4 compliant frame this will be just after the MSDU since the FCS is not reported. In the case of a non-complient frame it will be appended after the full frame. The LQI is a number ranging from 0 (lowest link quality) to 255 (highest link quality). The LQI is only valid for frames equal to or longer than three octets. When receiving a frame the RSSI (reported as negative dB) will be measured at three points during the reception. These three values will be sorted and the middle one selected (median 3) for then to be remapped within the LQI range. The following figure illustrates the LQI measurement and how the data is arranged in the DataRAM:

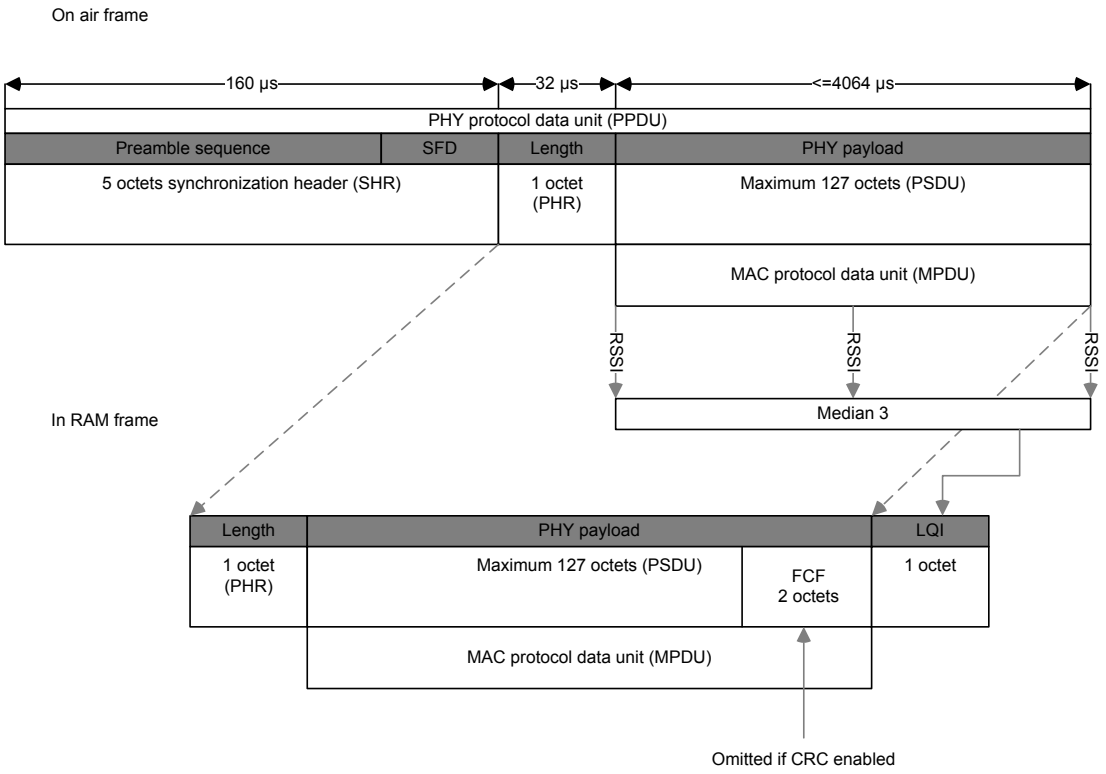

Figure 124: IEEE 802.15.4 frame in Data RAM

A shortcut has been added between FRAMESTART event and the BCSTART task. This can be used to trigger a BCMATCH event after N bits, such as when inspecting the MAC addressing fields.

6.20.12.8 Interframe spacing (IFS)

The IEEE 802.15.4 standard defines a specific time that is alotted for the MAC sublayer to process received data. Usage of this interframe spacing (IFS) comes into play to avoid that two frames are transmitted too close to eachother in time. If the a transmission is requesting an acknowledgement, the speration to the second frame shall be at least an IFS period.

The IFS is determined to be:

- IFS equals macMinSIFSPeriod (12 symbols) if the MPDU is less than or equal to aMaxSIFSFrameSize (18 octets) octets
- IFS equals macMinLIFSPeriod (40 symbols) if the MPDU is larger than aMaxSIFSFrameSize

Using the efficient assisted modes in the radio module the TIFS will be programmed with the correct value based on the frame being transmitted. If the assisted modes are not being used the user must update the TIFS register manually. The figure below provides details on what IFS period is valid in both acknowledged and unacknowledged transmissions.

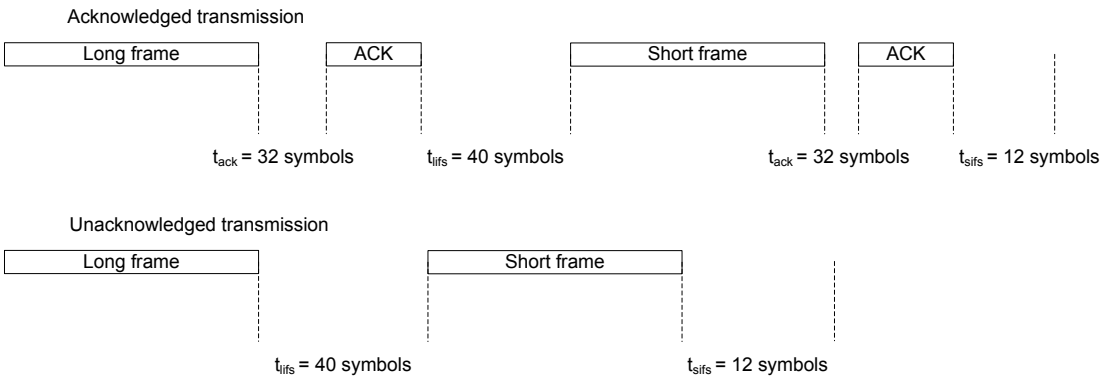

Figure 125: Interframe spacing examples

6.20.13 EasyDMA

The RADIO uses EasyDMA for reading of data packets from and writing to RAM, without CPU involvement.

As illustrated in RADIO block diagram on page 286, the RADIO's EasyDMA utilizes the same PACKETPTR for receiving and transmitting packets. This pointer should be reconfigured by the CPU each time before RADIO is started by the START task. The PACKETPTR registers is double-buffered, meaning that it can be updated and prepared for the next transmission.

Important: If the PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

The END event indicates that the last bit has been processed by the radio. The DISABLED event is issued to acknowledge that a DISABLE task is done.

The structure of a radio packet is described in detail in Packet configuration on page 286. The data that is stored in Data RAM and transported by EasyDMA consists of the following fields:

- S0
- LENGTH
- S1
- PAYLOAD

In addition, a static add-on is sent immediately after the payload.

The size of each of the above fields in the frame is configurable (see Packet configuration on page 286), and the space occupied in RAM depends on these settings. A size of zero is possible for any of the fields, it is up to the user to make sure that the resulting frame complies with the RF protocol chosen.

All fields are extended in size to align with a byte boundary in RAM. For instance a 3 bit long field on air will occupy 1 byte in RAM while a 9 bit long field will be extended to 2 bytes.

The radio packets elements can be configured as follows:

- CI, TERM1 and TERM2 fields are only present in *Bluetooth*® low energy long range mode
- S0 is configured through the S0LEN field in PCNF0
- LENGTH is configured through the LFLEN field in PCNF0
- S1 is configured through the S1LEN field in PCNF0
- Size of the payload is configured through the value in RAM corresponding to the LENGTH field
- Size of the static add-on to the payload is configured through the STATLEN field in PCNF1

The MAXLEN field in the PCNF1 register configures the maximum packet payload plus add-on size in number of bytes that can be transmitted or received by the RADIO. This feature can be used to ensure that the RADIO does not overwrite, or read beyond, the RAM assigned to the packet payload. This means

that if the packet payload length defined by PCNF1.STATLEN and the LENGTH field in the packet specifies a packet larger than MAXLEN, the payload will be truncated at MAXLEN.

Note: The MAXLEN includes the payload and the add-on, but excludes the size occupied by the S0, LENGTH and S1 fields. This has to be taken into account when allocating RAM.

If the payload and add-on length is specified larger than MAXLEN, the RADIO will still transmit or receive in the same way as before, except the payload is now truncated to MAXLEN. The packet's LENGTH field will not be altered when the payload is truncated. The RADIO will calculate CRC as if the packet length is equal to MAXLEN.

Note: If the PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

The END event indicates that the last bit has been processed by the radio. The DISABLED event is issued to acknowledge that an DISABLE task is done.

6.20.14 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40001000	RADIO	RADIO	2.4 GHz radio	

Table 85: Instances

Register	Offset	Description
TASKS_TXEN	0x000	Enable RADIO in TX mode
TASKS_RXEN	0x004	Enable RADIO in RX mode
TASKS_START	0x008	Start RADIO
TASKS_STOP	0x00C	Stop RADIO
TASKS_DISABLE	0x010	Disable RADIO
TASKS_RSSISTART	0x014	Start the RSSI and take one single sample of the receive signal strength
TASKS_RSSISTOP	0x018	Stop the RSSI measurement
TASKS_BCSTART	0x01C	Start the bit counter
TASKS_BCSTOP	0x020	Stop the bit counter
TASKS_EDSTART	0x024	Start the energy detect measurement used in IEEE 802.15.4 mode
TASKS_EDSTOP	0x028	Stop the energy detect measurement
TASKS_CCASTART	0x02C	Start the clear channel assessment used in IEEE 802.15.4 mode
TASKS_CCASTOP	0x030	Stop the clear channel assessment
EVENTS_READY	0x100	RADIO has ramped up and is ready to be started
EVENTS_ADDRESS	0x104	Address sent or received
EVENTS_PAYLOAD	0x108	Packet payload sent or received
EVENTS_END	0x10C	Packet sent or received
EVENTS_DISABLED	0x110	RADIO has been disabled
EVENTS_DEVMATCH	0x114	A device address match occurred on the last received packet
EVENTS_DEVMISS	0x118	No device address match occurred on the last received packet
EVENTS_RSSIEND	0x11C	Sampling of receive signal strength complete
EVENTS_BCMATCH	0x128	Bit counter reached bit count value
EVENTS_CRCOK	0x130	Packet received with CRC ok
EVENTS_CRCERROR	0x134	Packet received with CRC error
EVENTS_FRAMESTART	0x138	IEEE 802.15.4 length field received
EVENTS_EDEND	0x13C	Sampling of energy detection complete. A new ED sample is ready for readout from the RADIO.EDSAMPLE register.

Register	Offset	Description
EVENTS_EDSTOPPED	0x140	The sampling of energy detection has stopped
EVENTS_CCAIDLE	0x144	Wireless medium in idle - clear to send
EVENTS_CCABUSY	0x148	Wireless medium busy - do not send
EVENTS_CCASTOPPED	0x14C	The CCA has stopped
EVENTS_RATEBOOST	0x150	Ble_LR CI field received, receive mode is changed from Ble_LR125Kbit to Ble_LR500Kbit.
EVENTS_TXREADY	0x154	RADIO has ramped up and is ready to be started TX path
EVENTS_RXREADY	0x158	RADIO has ramped up and is ready to be started RX path
EVENTS_MHRMATCH	0x15C	MAC header match found
EVENTS_PHYEND	0x16C	Generated in Ble_LR125Kbit, Ble_LR500Kbit and BleIeee802154_250Kbit modes when last bit is sent on air.
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
CRCSTATUS	0x400	CRC status
RXMATCH	0x408	Received address
RXCRC	0x40C	CRC field of previously received packet
DAI	0x410	Device address match index
PDUSTAT	0x414	Payload status
PACKETPTR	0x504	Packet pointer
FREQUENCY	0x508	Frequency
TXPOWER	0x50C	Output power
MODE	0x510	Data rate and modulation
PCNF0	0x514	Packet configuration register 0
PCNF1	0x518	Packet configuration register 1
BASE0	0x51C	Base address 0
BASE1	0x520	Base address 1
PREFIX0	0x524	Prefixes bytes for logical addresses 0-3
PREFIX1	0x528	Prefixes bytes for logical addresses 4-7
TXADDRESS	0x52C	Transmit address select
RXADDRESSES	0x530	Receive address select
CRCCNF	0x534	CRC configuration
CRCPOLY	0x538	CRC polynomial
CRCINIT	0x53C	CRC initial value
TIFS	0x544	Interframe spacing in µs
RSSISAMPLE	0x548	RSSI sample
STATE	0x550	Current radio state
DATAWHITEIV	0x554	Data whitening initial value
BCC	0x560	Bit counter compare
DAB[0]	0x600	Device address base segment 0
DAB[1]	0x604	Device address base segment 1
DAB[2]	0x608	Device address base segment 2
DAB[3]	0x60C	Device address base segment 3
DAB[4]	0x610	Device address base segment 4
DAB[5]	0x614	Device address base segment 5
DAB[6]	0x618	Device address base segment 6
DAB[7]	0x61C	Device address base segment 7
DAP[0]	0x620	Device address prefix 0
DAP[1]	0x624	Device address prefix 1
DAP[2]	0x628	Device address prefix 2
DAP[3]	0x62C	Device address prefix 3
DAP[4]	0x630	Device address prefix 4
DAP[5]	0x634	Device address prefix 5
DAP[6]	0x638	Device address prefix 6

Register	Offset	Description
DAP[7]	0x63C	Device address prefix 7
DACNF	0x640	Device address match configuration
MHRMATCHCONF	0x644	Search pattern configuration
MHRMATCHMAS	0x648	Pattern mask
MODECNF0	0x650	Radio mode configuration register 0
SFD	0x660	IEEE 802.15.4 start of frame delimiter
EDCNT	0x664	IEEE 802.15.4 energy detect loop count
EDSAMPLE	0x668	IEEE 802.15.4 energy detect level
CCACTRL	0x66C	IEEE 802.15.4 clear channel assessment control
POWER	0xFFC	Peripheral power control

Table 86: Register overview

6.20.14.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID															U	T	S	R	Q	P	O	N	M	L	K			H		G	F	E	D	C	B	A
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value	Description																															
A	RW	READY_START			Shortcut between READY event and START task See EVENTS_READY and TASKS_START																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
B	RW	END_DISABLE			Shortcut between END event and DISABLE task See EVENTS_END and TASKS_DISABLE																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
C	RW	DISABLED_TXEN			Shortcut between DISABLED event and TXEN task See EVENTS_DISABLED and TASKS_TXEN																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
D	RW	DISABLED_RXEN			Shortcut between DISABLED event and RXEN task See EVENTS_DISABLED and TASKS_RXEN																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
E	RW	ADDRESS_RSSISTART			Shortcut between ADDRESS event and RSSISTART task See EVENTS_ADDRESS and TASKS_RSSISTART																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
F	RW	END_START			Shortcut between END event and START task See EVENTS_END and TASKS_START																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																
G	RW	ADDRESS_BCSTART			Shortcut between ADDRESS event and BCSTART task See EVENTS_ADDRESS and TASKS_BCSTART																																
			Disabled	0	Disable shortcut																																
			Enabled	1	Enable shortcut																																

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				U T S R Q P O N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
H	RW	DISABLED_RSSISTOP			Shortcut between DISABLED event and RSSISTOP task See EVENTS_DISABLED and TASKS_RSSISTOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
K	RW	RXREADY_CCASTART			Shortcut between RXREADY event and CCASTART task See EVENTS_RXREADY and TASKS_CCASTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
L	RW	CCAIDLE_TXEN			Shortcut between CCAIDLE event and TXEN task See EVENTS_CCAIDLE and TASKS_TXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
M	RW	CCABUSY_DISABLE			Shortcut between CCABUSY event and DISABLE task See EVENTS_CCABUSY and TASKS_DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
N	RW	FRAMESTART_BCSTART			Shortcut between FRAMESTART event and BCSTART task See EVENTS_FRAMESTART and TASKS_BCSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
O	RW	READY_EDSTART			Shortcut between READY event and EDSTART task See EVENTS_READY and TASKS_EDSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
P	RW	EDEND_DISABLE			Shortcut between EDEND event and DISABLE task See EVENTS_EDEND and TASKS_DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
Q	RW	CCAIDLE_STOP			Shortcut between CCAIDLE event and STOP task See EVENTS_CCAIDLE and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
R	RW	TXREADY_START			Shortcut between TXREADY event and START task See EVENTS_TXREADY and TASKS_START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
S	RW	RXREADY_START			Shortcut between RXREADY event and START task See EVENTS_RXREADY and TASKS_START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
T	RW	PHYEND_DISABLE			Shortcut between PHYEND event and DISABLE task See EVENTS_PHYEND and TASKS_DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
U	RW	PHYEND_START			Shortcut between PHYEND event and START task See EVENTS_PHYEND and TASKS_START

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	U T S R Q P O N M L K H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.20.14.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	Z V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ADDRESS			Write '1' to enable interrupt for ADDRESS event See EVENTS_ADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	PAYLOAD			Write '1' to enable interrupt for PAYLOAD event See EVENTS_PAYLOAD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	DISABLED			Write '1' to enable interrupt for DISABLED event See EVENTS_DISABLED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	DEVMATCH			Write '1' to enable interrupt for DEVMATCH event See EVENTS_DEVMATCH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	DEVMISS			Write '1' to enable interrupt for DEVMISS event See EVENTS_DEVMISS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID					Z				V	U	T	S	R	Q	P	O	N	M	L	K		I			H	G	F	E	D	C	B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
H	RW	RSSIEND			Write '1' to enable interrupt for RSSIEND event See EVENTS_RSSIEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	BCMATCH			Write '1' to enable interrupt for BCMATCH event See EVENTS_BCMATCH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CRCOK			Write '1' to enable interrupt for CRCOK event See EVENTS_CRCOK
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CRCERROR			Write '1' to enable interrupt for CRCERROR event See EVENTS_CRCERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	FRAMESTART			Write '1' to enable interrupt for FRAMESTART event See EVENTS_FRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	EDEND			Write '1' to enable interrupt for EDEND event See EVENTS_EDEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	EDSTOPPED			Write '1' to enable interrupt for EDSTOPPED event See EVENTS_EDSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CCAIDLE			Write '1' to enable interrupt for CCAIDLE event See EVENTS_CCAIDLE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CCABUSY			Write '1' to enable interrupt for CCABUSY event See EVENTS_CCABUSY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CCASTOPPED			Write '1' to enable interrupt for CCASTOPPED event See EVENTS_CCASTOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Z V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	RATEBOOST			Write '1' to enable interrupt for RATEBOOST event See EVENTS_RATEBOOST
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	TXREADY			Write '1' to enable interrupt for TXREADY event See EVENTS_TXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	RXREADY			Write '1' to enable interrupt for RXREADY event See EVENTS_RXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	MHRMATCH			Write '1' to enable interrupt for MHRMATCH event See EVENTS_MHRMATCH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Z	RW	PHYEND			Write '1' to enable interrupt for PHYEND event See EVENTS_PHYEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.20.14.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Z V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ADDRESS			Write '1' to disable interrupt for ADDRESS event See EVENTS_ADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID									Z				V	U	T	S	R	Q	P	O	N	M	L	K		I			H	G	F	E	D	C	B	A
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID		Value								Description																							
			Enabled		1								Read: Enabled																							
C	RW	PAYLOAD											Write '1' to disable interrupt for PAYLOAD event See EVENTS_PAYLOAD																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
D	RW	END											Write '1' to disable interrupt for END event See EVENTS_END																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
E	RW	DISABLED											Write '1' to disable interrupt for DISABLED event See EVENTS_DISABLED																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
F	RW	DEVMATCH											Write '1' to disable interrupt for DEVMATCH event See EVENTS_DEVMATCH																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
G	RW	DEVMISS											Write '1' to disable interrupt for DEVMISS event See EVENTS_DEVMISS																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
H	RW	RSSIEND											Write '1' to disable interrupt for RSSIEND event See EVENTS_RSSIEND																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
I	RW	BCMATCH											Write '1' to disable interrupt for BCMATCH event See EVENTS_BCMATCH																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
K	RW	CRCOK											Write '1' to disable interrupt for CRCOK event See EVENTS_CRCOK																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							
L	RW	CRCERROR											Write '1' to disable interrupt for CRCERROR event See EVENTS_CRCERROR																							
			Clear		1								Disable																							
			Disabled		0								Read: Disabled																							
			Enabled		1								Read: Enabled																							

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Z V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
M	RW	FRAMESTART			Write '1' to disable interrupt for FRAMESTART event See EVENTS_FRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	EDEND			Write '1' to disable interrupt for EDEND event See EVENTS_EDEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	EDSTOPPED			Write '1' to disable interrupt for EDSTOPPED event See EVENTS_EDSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CCAIDLE			Write '1' to disable interrupt for CCAIDLE event See EVENTS_CCAIDLE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CCABUSY			Write '1' to disable interrupt for CCABUSY event See EVENTS_CCABUSY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CCASTOPPED			Write '1' to disable interrupt for CCASTOPPED event See EVENTS_CCASTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	RATEBOOST			Write '1' to disable interrupt for RATEBOOST event See EVENTS_RATEBOOST
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	TXREADY			Write '1' to disable interrupt for TXREADY event See EVENTS_TXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	RXREADY			Write '1' to disable interrupt for RXREADY event See EVENTS_RXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	MHRMATCH			Write '1' to disable interrupt for MHRMATCH event See EVENTS_MHRMATCH

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Z V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Z	RW	PHYEND			Write '1' to disable interrupt for PHYEND event See EVENTS_PHYEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.20.14.4 CRCSTATUS

Address offset: 0x400

CRC status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	CRCSTATUS			CRC status of packet received
			CRCError	0	Packet received with CRC error
			CRCOk	1	Packet received with CRC ok

6.20.14.5 RXMATCH

Address offset: 0x408

Received address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXMATCH			Received address Logical address of which previous packet was received

6.20.14.6 RXCRC

Address offset: 0x40C

CRC field of previously received packet

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXCRC			CRC field of previously received packet CRC field of previously received packet

6.20.14.7 DAI

Address offset: 0x410

Device address match index

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	DAI			Device address match index
					Index (n) of device address, see DAB[n] and DAP[n], that got an address match

6.20.14.8 PDUSTAT

Address offset: 0x414

Payload status

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	B B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	PDUSTAT			Status on payload length vs. PCNF1.MAXLEN
			LessThan	0	Payload less than PCNF1.MAXLEN
			GreaterThan	1	Payload greater than PCNF1.MAXLEN
B	R	CISTAT			Status on what rate packet is received with in Long Range
			LR125kbit	0	Frame is received at 125kbps
			LR500kbit	1	Frame is received at 500kbps

6.20.14.9 PACKETPTR

Address offset: 0x504

Packet pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PACKETPTR			Packet pointer
					Packet address to be used for the next transmission or reception. When transmitting, the packet pointed to by this address will be transmitted and when receiving, the received packet will be written to this address. This address is a byte aligned RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.20.14.10 FREQUENCY

Address offset: 0x508

Frequency

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQUENCY		[0..100]	Radio channel frequency Frequency = 2400 + FREQUENCY (MHz).
B	RW	MAP			Channel map selection.
			Default	0	Channel map between 2400 MHZ .. 2500 MHz Frequency = 2400 + FREQUENCY (MHz)
			Low	1	Channel map between 2360 MHZ .. 2460 MHz Frequency = 2360 + FREQUENCY (MHz)

6.20.14.11 TXPOWER

Address offset: 0x50C

Output power

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID				A A A A A A A A	
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**	
ID	RW	Field	Value ID	Value	Description
A	RW	TXPOWER			RADIO output power Output power in number of dBm, i.e. if the value -20 is specified the output power will be set to -20dBm.
			Pos8dBm	0x8	+8 dBm
			Pos7dBm	0x7	+7 dBm
			Pos6dBm	0x6	+6 dBm
			Pos5dBm	0x5	+5 dBm
			Pos4dBm	0x4	+4 dBm
			Pos3dBm	0x3	+3 dBm
			Pos2dBm	0x2	+2 dBm
			0dBm	0x0	0 dBm
			Neg4dBm	0xFC	-4 dBm
			Neg8dBm	0xF8	-8 dBm
			Neg12dBm	0xF4	-12 dBm
			Neg16dBm	0xF0	-16 dBm
			Neg20dBm	0xEC	-20 dBm
			Neg30dBm	0xFF	-40 dBm Deprecated
			Neg40dBm	0xD8	-40 dBm

6.20.14.12 MODE

Address offset: 0x510

Data rate and modulation

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MODE			Radio data rate and modulation setting. The radio supports frequency-shift keying (FSK) modulation.
			Nrf_1Mbit	0	1 Mbit/s Nordic proprietary radio mode
			Nrf_2Mbit	1	2 Mbit/s Nordic proprietary radio mode
			Ble_1Mbit	3	1 Mbit/s BLE
			Ble_2Mbit	4	2 Mbit/s BLE
			Ble_LR125Kbit	5	Long range 125 kbit/s TX, 125 kbit/s and 500 kbit/s RX
			Ble_LR500Kbit	6	Long range 500 kbit/s TX, 125 kbit/s and 500 kbit/s RX
			Ieee802154_250Kbit	15	IEEE 802.15.4-2006 250 kbit/s

6.20.14.13 PCNF0

Address offset: 0x514

Packet configuration register 0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J J I H H G G F E E E E C A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LFLEN			Length on air of LENGTH field in number of bits.
C	RW	S0LEN			Length on air of S0 field in number of bytes.
E	RW	S1LEN			Length on air of S1 field in number of bits.
F	RW	S1INCL			Include or exclude S1 field in RAM
			Automatic	0	Include S1 field in RAM only if S1LEN > 0
			Include	1	Always include S1 field in RAM independent of S1LEN
G	RW	CILEN			Length of code indicator - long range
H	RW	PLEN			Length of preamble on air. Decision point: TASKS_START task
			8bit	0	8-bit preamble
			16bit	1	16-bit preamble
			32bitZero	2	32-bit zero preamble - used for IEEE 802.15.4
			LongRange	3	Preamble - used for BLE long range
I	RW	CRCINC			Indicates if LENGTH field contains CRC or not
			Exclude	0	LENGTH does not contain CRC
			Include	1	LENGTH includes CRC
J	RW	TERMLEN			Length of TERM field in Long Range operation

6.20.14.14 PCNF1

Address offset: 0x518

Packet configuration register 1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C C C B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXLEN		[0..255]	Maximum length of packet payload. If the packet payload is larger than MAXLEN, the radio will truncate the payload to MAXLEN.
B	RW	STATLEN		[0..255]	Static length in number of bytes The static length parameter is added to the total length of the payload when sending and receiving packets, e.g. if the static length is set to N the radio will receive or send N bytes more than what is defined in the LENGTH field of the packet.
C	RW	BALEN		[2..4]	Base address length in number of bytes The address field is composed of the base address and the one byte long address prefix, e.g. set BALEN=2 to get a total address of 3 bytes.
D	RW	ENDIAN			On air endianness of packet, this applies to the S0, LENGTH, S1 and the PAYLOAD fields.
			Little	0	Least significant bit on air first
			Big	1	Most significant bit on air first
E	RW	WHITEEN			Enable or disable packet whitening
			Disabled	0	Disable
			Enabled	1	Enable

6.20.14.15 BASE0

Address offset: 0x51C

Base address 0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	BASE0			Base address 0 Radio base address 0.

6.20.14.16 BASE1

Address offset: 0x520

Base address 1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	BASE1			Base address 1 Radio base address 1.

6.20.14.17 PREFIX0

Address offset: 0x524

Prefixes bytes for logical addresses 0-3

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-D	RW	AP[i] (i=0..3)			Address prefix i.

6.20.14.18 PREFIX1

Address offset: 0x528

Prefixes bytes for logical addresses 4-7

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-D	RW	AP[i] (i=4..7)			Address prefix i.

6.20.14.19 TXADDRESS

Address offset: 0x52C

Transmit address select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TXADDRESS			Transmit address select Logical address to be used when transmitting a packet.

6.20.14.20 RXADDRESSES

Address offset: 0x530

Receive address select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-H	RW	ADDR[i] (i=0..7)			Enable or disable reception on logical address i.
			Disabled	0	Disable
			Enabled	1	Enable

6.20.14.21 CRCCNF

Address offset: 0x534

CRC configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LEN		[1..3]	CRC length in number of bytes. **Note:** For MODE Ble_LR125Kbit and Ble_LR500Kbit, only LEN set to 3 is supported
			Disabled	0	CRC length is zero and CRC calculation is disabled
			One	1	CRC length is one byte and CRC calculation is enabled
			Two	2	CRC length is two bytes and CRC calculation is enabled
			Three	3	CRC length is three bytes and CRC calculation is enabled
B	RW	SKIPADDR			Include or exclude packet address field out of CRC calculation.
			Include	0	CRC calculation includes address field
			Skip	1	CRC calculation does not include address field. The CRC calculation will start at the first byte after the address.
			Ieee802154	2	CRC calculation as per 802.15.4 standard. Starting at first byte after length field.

6.20.14.22 CRCPOLY

Address offset: 0x538

CRC polynomial

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CRCPOLY			CRC polynomial Each term in the CRC polynomial is mapped to a bit in this register which index corresponds to the term's exponent. The least significant term/bit is hard-wired internally to 1, and bit number 0 of the register content is ignored by the hardware. The following example is for an 8 bit CRC polynomial: x8 + x7 + x3 + x2 + 1 = 1 1000 1101 .

6.20.14.23 CRCINIT

Address offset: 0x53C

CRC initial value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CRCINIT			CRC initial value Initial value for CRC calculation

6.20.14.24 TIFS

Address offset: 0x544

Interframe spacing in µs

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	TIFS			Interframe spacing in µs Interframe space is the time interval between two consecutive packets. It is defined as the time, in microseconds, from the end of the last bit of the previous packet to the start of the first bit of the subsequent packet.

6.20.14.25 RSSISAMPLE

Address offset: 0x548

RSSI sample

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	RSSISAMPLE		[0..127]	RSSI sample RSSI sample result. The value of this register is read as a positive value while the actual received signal strength is a negative value. Actual received signal strength is therefore as follows: received signal strength = -A dBm

6.20.14.26 STATE

Address offset: 0x550

Current radio state

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	STATE			Current radio state
			Disabled	0	RADIO is in the Disabled state
			RxRu	1	RADIO is in the RXRU state
			RxIdle	2	RADIO is in the RXIDLE state
			Rx	3	RADIO is in the RX state
			RxDisable	4	RADIO is in the RXDISABLED state
			TxRu	9	RADIO is in the TXRU state
			TxIdle	10	RADIO is in the TXIDLE state
			Tx	11	RADIO is in the TX state
			TxDisable	12	RADIO is in the TXDISABLED state

6.20.14.27 DATAWHITEIV

Address offset: 0x554

Data whitening initial value

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A
Reset 0x00000040	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DATAWHITEIV			Data whitening initial value. Bit 6 is hard-wired to '1', writing '0' to it has no effect, and it will always be read back and used by the device as '1'. Bit 0 corresponds to Position 6 of the LSFR, Bit 1 to Position 5, etc.

6.20.14.28 BCC

Address offset: 0x560

Bit counter compare

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	BCC			Bit counter compare Bit counter compare register

6.20.14.29 DAB[n] (n=0..7)

Address offset: 0x600 + (n × 0x4)

Device address base segment n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DAB			Device address base segment n

6.20.14.30 DAP[n] (n=0..7)

Address offset: 0x620 + (n × 0x4)

Device address prefix n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DAP			Device address prefix n

6.20.14.31 DACNF

Address offset: 0x640

Device address match configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-H	RW	ENA[i] (i=0..7)			Enable or disable device address matching using device address i
			Disabled	0	Disabled
			Enabled	1	Enabled
I-P	RW	TXADD[i] (i=0..7)			TxAdd for device address i

6.20.14.32 MHRMATCHCONF

Address offset: 0x644

Search pattern configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description

6.20.14.33 MHRMATCHMAS

Address offset: 0x648

Pattern mask

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description

6.20.14.34 MODECNF0

Address offset: 0x650

Radio mode configuration register 0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C A
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RU			Radio ramp-up time
			Default	0	Default ramp-up time (tRXEN), compatible with firmware written for nRF51
			Fast	1	Fast ramp-up (tRXEN,FAST), see electrical specification for more information

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C A
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	DTX			Default TX value Specifies what the RADIO will transmit when it is not started, i.e. between: RADIO.EVENTS_READY and RADIO.TASKS_START RADIO.EVENTS_END and RADIO.TASKS_START RADIO.EVENTS_END and RADIO.EVENTS_DISABLED **Note:** For 802.15.4 and BLE LR mode, only Center is a valid setting
			B1	0	Transmit '1'
			B0	1	Transmit '0'
			Center	2	Transmit center frequency When tuning the crystal for centre frequency, the RADIO must be set in DTX = Center mode to be able to achieve the expected accuracy

6.20.14.35 SFD

Address offset: 0x660

IEEE 802.15.4 start of frame delimiter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x000000A7				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	SFD			IEEE 802.15.4 start of frame delimiter

6.20.14.36 EDCNT

Address offset: 0x664

IEEE 802.15.4 energy detect loop count

Number of iterations to perform an ED scan. If set to 0 one scan is performed, otherwise the specified number + 1 of ED scans will be performed and the max ED value tracked in EDSAMPLE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EDCNT			IEEE 802.15.4 energy detect loop count

6.20.14.37 EDSAMPLE

Address offset: 0x668

IEEE 802.15.4 energy detect level

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EDLVL			IEEE 802.15.4 energy detect level

6.20.14.38 CCACTRL

Address offset: 0x66C

IEEE 802.15.4 clear channel assessment control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CCAMODE			CCA mode of operation
			EdMode	0	Energy above threshold Will report busy whenever energy is detected above CCAEDTHRES
			CarrierMode	1	Carrier seen Will report busy whenever compliant IEEE 802.15.4 signal is seen
			CarrierAndEdMode	2	Energy above threshold AND carrier seen
			CarrierOrEdMode	3	Energy above threshold OR carrier seen
			EdModeTest1	4	Energy above threshold test mode that will abort when first ED measurement over threshold is seen. No averaging.
B	RW	CCAEDTHRES			CCA energy busy threshold. Used in all the CCA modes except CarrierMode.
C	RW	CCACORRTHRES			CCA correlator busy threshold. Only relevant to CarrierMode, CarrierAndEdMode and CarrierOrEdMode.
D	RW	CCACORRCNT			Limit for occurances above CCACORRTHRES. When not equal to zero the corrolator based signal detect is enabled.

6.20.14.39 POWER

Address offset: 0xFFC

Peripheral power control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	POWER			Peripheral power control. The peripheral and its registers will be reset to its initial state by switching the peripheral off and then back on again.
			Disabled	0	Peripheral is powered off
			Enabled	1	Peripheral is powered on

6.20.15 Electrical specification

6.20.15.1 General radio characteristics

Symbol	Description	Min.	Typ.	Max.	Units
f_{OP}	Operating frequencies	2360		2500	MHz
$f_{PLL,CH,SP}$	PLL channel spacing		1		MHz
$f_{DELTA,1M}$	Frequency deviation @ 1 Mbps		±170		kHz
$f_{DELTA,BLE,1M}$	Frequency deviation @ BLE 1 Mbps		±250		kHz
$f_{DELTA,2M}$	Frequency deviation @ 2 Mbps		±320		kHz
$f_{DELTA,BLE,2M}$	Frequency deviation @ BLE 2 Mbps		±500		kHz
fsk_{BPS}	On the air data rate	125		2000	kbps
$f_{chip,\ IEEE\ 802.15.4}$	Chip rate in IEEE 802.15.4 mode		2000		kchip/s

6.20.15.2 Radio current consumption (transmitter)

Symbol	Description	Min.	Typ.	Max.	Units
$I_{TX,PLUS8dBM,DCDC}$	TX only run current (DC/DC, 3 V) P_{RF} = +8 dBm	..	14.8	..	mA
$I_{TX,PLUS8dBM}$	TX only run current P_{RF} = +8 dBm	..	32.7	..	mA
$I_{TX,PLUS4dBM,DCDC}$	TX only run current (DC/DC, 3 V) P_{RF} = +4 dBm	..	9.6	..	mA
$I_{TX,PLUS4dBM}$	TX only run current P_{RF} = +4 dBm	..	21.4	..	mA
$I_{TX,0dBM,DCDC,5V,REG0HIGH}$	TX only run current (DC/DC, 5 V, REG0 out = 3.3 V)P_{RF} = 0 dBm		3.0		mA
$I_{TX,0dBM,DCDC,5V,REG0L}$	TX only run current (DC/DC, 5 V, REG0 out = 1.8 V)P_{RF} = 0 dBm		3.0		mA
$I_{TX,0dBM,DCDC}$	TX only run current (DC/DC, 3 V)P_{RF} = 0 dBm	..	4.8	8.7	mA
$I_{TX,0dBM}$	TX only run current P_{RF} = 0 dBm	..	10.6	..	mA
$I_{TX,MINUS4dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -4 dBm	..	3.1	..	mA
$I_{TX,MINUS4dBM}$	TX only run current P_{RF} = -4 dBm	..	8.1	..	mA
$I_{TX,MINUS8dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -8 dBm		3.3		mA
$I_{TX,MINUS8dBM}$	TX only run current P_{RF} = -8 dBm	..	7.2	7.9	mA
$I_{TX,MINUS12dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -12 dBm		3.0		mA
$I_{TX,MINUS12dBM}$	TX only run current P_{RF} = -12 dBm	..	6.4	..	mA
$I_{TX,MINUS16dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -16 dBm		2.8		mA
$I_{TX,MINUS16dBM}$	TX only run current P_{RF} = -16 dBm	..	6.0	..	mA
$I_{TX,MINUS20dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -20 dBm		2.7		mA
$I_{TX,MINUS20dBM}$	TX only run current P_{RF} = -20 dBm	..	5.6	..	mA
$I_{TX,MINUS40dBM,DCDC}$	TX only run current DC/DC, 3 V P_{RF} = -40 dBm		2.3		mA
$I_{TX,MINUS40dBM}$	TX only run current P_{RF} = -40 dBm	..	4.6	..	mA
$I_{START,TX,DCDC}$	TX start-up current DC/DC, 3 V, P_{RF} = 4 dBm		5.2		mA
$I_{START,TX}$	TX start-up current, P_{RF} = 4 dBm		11.0		mA

6.20.15.3 Radio current consumption (Receiver)

Symbol	Description	Min.	Typ.	Max.	Units
$I_{RX,1M,DCDC}$	RX only run current (DC/DC, 3 V) 1 Mbps / 1 Mbps BLE	..	4.6	..	mA
$I_{RX,1M}$	RX only run current (LDO, 3 V) 1 Mbps / 1 Mbps BLE	..	9.9	..	mA
$I_{RX,2M,DCDC}$	RX only run current (DC/DC, 3 V) 2 Mbps / 2 Mbps BLE	..	5.2	..	mA
$I_{RX,2M}$	RX only run current (LDO, 3 V) 2 Mbps / 2 Mbps BLE	..	11.1	..	mA
$I_{START,RX,1M,DCDC}$	RX start-up current (DC/DC, 3 V) 1 Mbps / 1 Mbps BLE		3.7		mA
$I_{START,RX,1M}$	RX start-up current 1 Mbps / 1 Mbps BLE		6.7		mA

6.20.15.4 Transmitter specification

Symbol	Description	Min.	Typ.	Max.	Units
P_{RF}	Maximum output power	..	8.0	..	dBm
P_{RFC}	RF power control range		28.0		dB
P_{RFCR}	RF power accuracy			±4	dB
$P_{RF1,1}$	1st Adjacent Channel Transmit Power 1 MHz (1 Mbps)	..	-24.8	..	dBc
$P_{RF2,1}$	2nd Adjacent Channel Transmit Power 2 MHz (1 Mbps)	..	-54.0	..	dBc
$P_{RF1,2}$	1st Adjacent Channel Transmit Power 2 MHz (2 Mbps)	..	-25	..	dBc
$P_{RF2,2}$	2nd Adjacent Channel Transmit Power 4 MHz (2 Mbps)	..	-54.0	..	dBc
E_{vm}	Error vector magnitude IEEE 802.15.4	..	8	..	%rms
$P_{harm2nd, IEEE 802.15.4}$	2nd harmonics in IEEE 802.15.4 mode	..	-51.0	..	dBm
$P_{harm3rd, IEEE 802.15.4}$	3rd harmonics in IEEE 802.15.4		-48.0	..	dBm

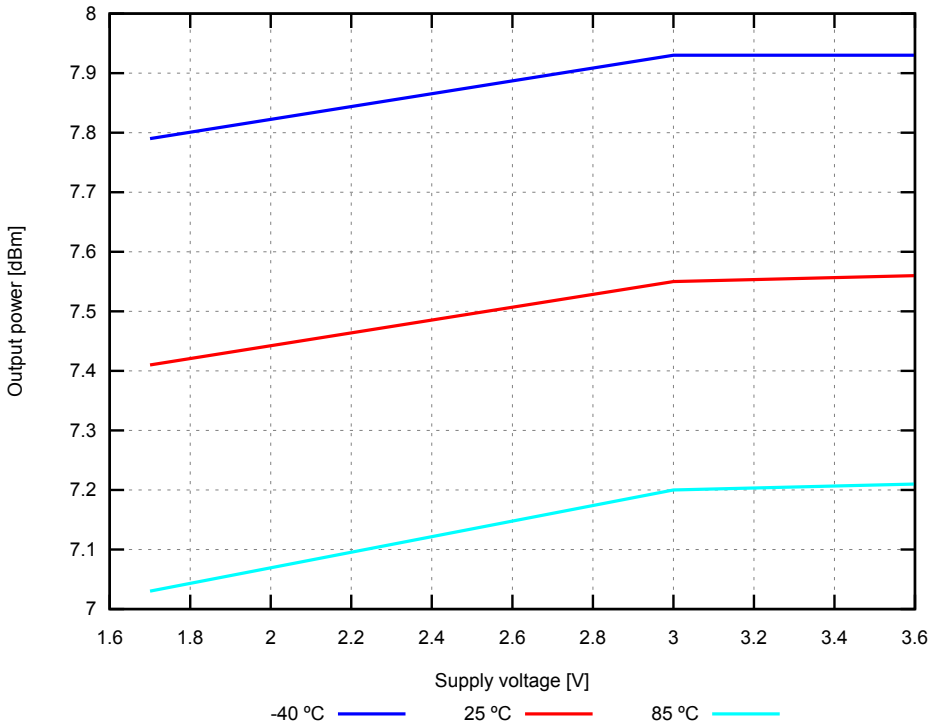

Figure 126: Output power, 1 Mbps Bluetooth low energy mode, 8 dBm TXPOWER setting (typical values)

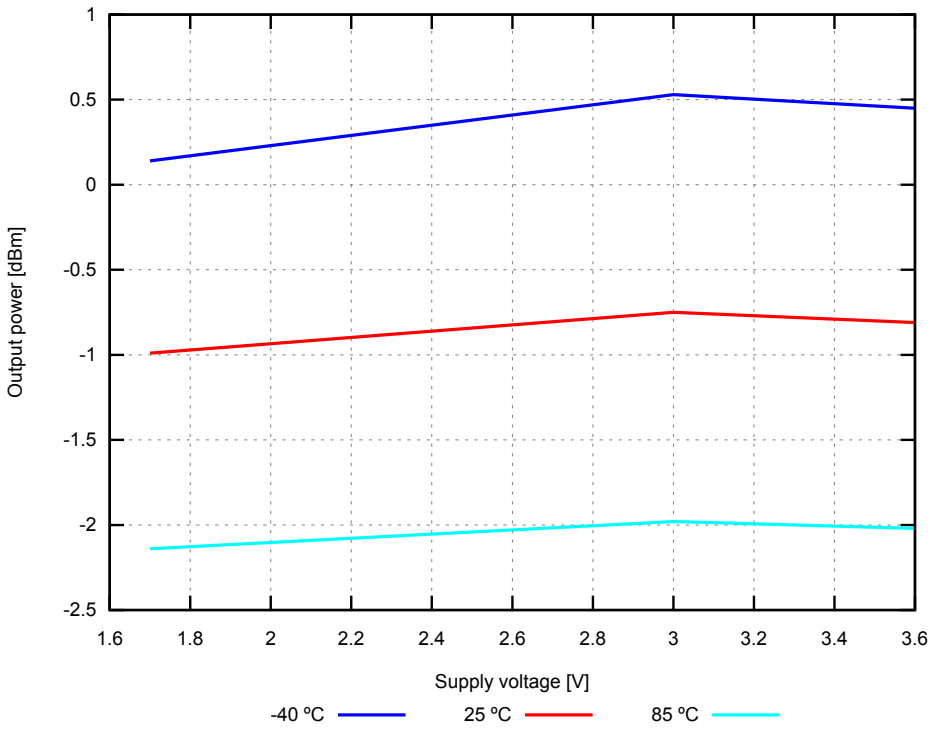

Figure 127: Output power, 1 Mbps Bluetooth low energy mode, 0 dBm TXPOWER setting (typical values)

6.20.15.5 Receiver operation

Symbol	Description	Min.	Typ.	Max.	Units
$P_{RX,MAX}$	Maximum received signal strength at < 0.1% PER		0		dBm
$P_{SENS,IT,1M}$	Sensitivity, 1 Mbps nRF mode[20]		-93		dBm
$P_{SENS,IT,SP,1M,BLE}$	Sensitivity, 1 Mbps BLE ideal transmitter, <=37 bytes BER=1E-3[21]		-95		dBm
$P_{SENS,IT,LP,1M,BLE}$	Sensitivity, 1 Mbps BLE ideal transmitter >=128 bytes BER=1E-4 [22]		-94.0		dBm
$P_{SENS,IT,2M}$	Sensitivity, 2 Mbps nRF mode[23]				dBm
$P_{SENS,IT,SP,2M,BLE}$	Sensitivity, 2 Mbps BLE ideal transmitter, Packet length <=37 bytes		-92		dBm
$P_{SENS,IT,BLE\ LE125k}$	Sensitivity, 125 kbps BLE mode		-103.0		dBm
$P_{SENS,IT,BLE\ LE500k}$	Sensitivity, 500 kbps BLE mode		-99		dBm
$P_{sense,\ IEEE\ 802.15.4}$	Sensitivity in IEEE 802.15.4 mode		-100		dBm

[20] Typical sensitivity applies when ADDR0 is used for receiver address correlation. When ADDR[1...7] are used for receiver address correlation, the typical sensitivity for this mode is degraded by 3 dB.

[21] As defined in the Bluetooth Core Specification v4.0 Volume 6: Core System Package (Low Energy Controller Volume)

[22] Equivalent BER limit < 10E-04

[23] Typical sensitivity applies when ADDR0 is used for receiver address correlation. When ADDR[1...7] are used for receiver address correlation, the typical sensitivity for this mode is degraded by 3 dB.

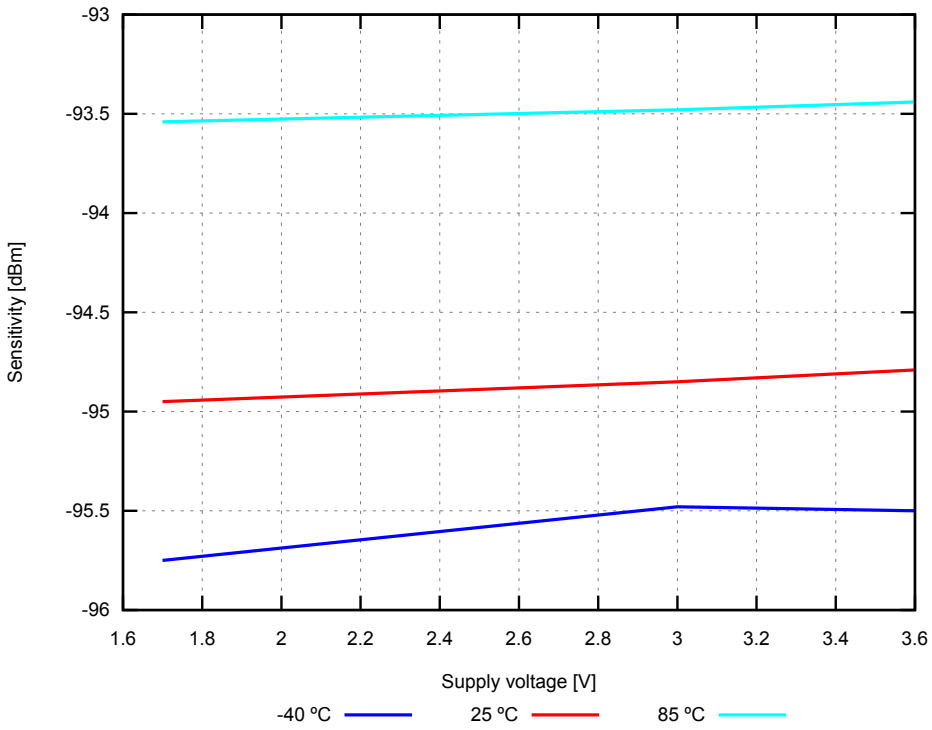

Figure 128: Sensitivity, 1 Mbps Bluetooth low energy mode, Regulator = LDO (typical values)

6.20.15.6 RX selectivity

RX selectivity with equal modulation on interfering signal[24]

Symbol	Description	Min.	Typ.	Max.	Units
$C/I_{1M,co\text{-}channel}$	1Mbps mode, Co-Channel interference		9		dB
$C/I_{1M,-1MHz}$	1 Mbps mode, Adjacent (-1 MHz) interference		-2		dB
$C/I_{1M,+1MHz}$	1 Mbps mode, Adjacent (+1 MHz) interference		-10		dB
$C/I_{1M,-2MHz}$	1 Mbps mode, Adjacent (-2 MHz) interference		-19		dB
$C/I_{1M,+2MHz}$	1 Mbps mode, Adjacent (+2 MHz) interference		-42		dB
$C/I_{1M,-3MHz}$	1 Mbps mode, Adjacent (-3 MHz) interference		-38		dB
$C/I_{1M,+3MHz}$	1 Mbps mode, Adjacent (+3 MHz) interference		-48		dB
$C/I_{1M,\pm6MHz}$	1 Mbps mode, Adjacent (≥6 MHz) interference		-50		dB
$C/I_{1MBLE,co\text{-}channel}$	1 Mbps BLE mode, Co-Channel interference		6		dB
$C/I_{1MBLE,-1MHz}$	1 Mbps BLE mode, Adjacent (-1 MHz) interference		-2		dB
$C/I_{1MBLE,+1MHz}$	1 Mbps BLE mode, Adjacent (+1 MHz) interference		-9		dB
$C/I_{1MBLE,-2MHz}$	1 Mbps BLE mode, Adjacent (-2 MHz) interference		-22		dB
$C/I_{1MBLE,+2MHz}$	1 Mbps BLE mode, Adjacent (+2 MHz) interference		-46		dB
$C/I_{1MBLE,>3MHz}$	1 Mbps BLE mode, Adjacent (≥3 MHz) interference		-50		dB
$C/I_{1MBLE,image}$	Image frequency interference		-22		dB
$C/I_{1MBLE,image,1MHz}$	Adjacent (1 MHz) interference to in-band image frequency		-35		dB
$C/I_{2M,co\text{-}channel}$	2 Mbps mode, Co-Channel interference		10		dB
$C/I_{2M,-2MHz}$	2 Mbps mode, Adjacent (-2 MHz) interference		6		dB
$C/I_{2M,+2MHz}$	2 Mbps mode, Adjacent (+2 MHz) interference		-19		dB
$C/I_{2M,-4MHz}$	2 Mbps mode, Adjacent (-4 MHz) interference		-20		dB
$C/I_{2M,+4MHz}$	2 Mbps mode, Adjacent (+4 MHz) interference		-44		dB

[24] Wanted signal level at PIN = -67 dBm. One interferer is used, having equal modulation as the wanted signal. The input power of the interferer where the sensitivity equals BER = 0.1% is presented

Symbol	Description	Min.	Typ.	Max.	Units
$C/I_{2M,-6MHz}$	2 Mbps mode, Adjacent (-6 MHz) interference		-42		dB
$C/I_{2M,+6MHz}$	2 Mbps mode, Adjacent (+6 MHz) interference		-42		dB
$C/I_{2M,\geq12MHz}$	2 Mbps mode, Adjacent (≥12 MHz) interference		-52		dB
$C/I_{2MBLE,co\text{-}channel}$	2 Mbps BLE mode, Co-Channel interference		6.8		dB
$C/I_{2MBLE,\pm2MHz}$	2 Mbps BLE mode, Adjacent (±2 MHz) interference		-10		dB
$C/I_{2MBLE,\pm4MHz}$	2 Mbps BLE mode, Adjacent (±4 MHz) interference		-45		dB
$C/I_{2MBLE,\geq6MHz}$	2 Mbps BLE mode, Adjacent (≥6 MHz) interference		-48		dB
$C/I_{2MBLE,image}$	Image frequency interference		-24		dB
$C/I_{2MBLE,image,\ 2MHz}$	Adjacent (2 MHz) interference to in-band image frequency		-35		dB
$C/I_{125k\ BLE\ LR,co\text{-}channel}$	125 kbps BLE LR mode, Co-Channel interference		4.4		dB
$C/I_{125k\ BLE\ LR,-1MHz}$	125 kbps BLE LR mode, Adjacent (-1 MHz) interference		-4.0		dB
$C/I_{125k\ BLE\ LR,+1MHz}$	125 kbps BLE LR mode, Adjacent (+1 MHz) interference		-12		dB
$C/I_{125k\ BLE\ LR,-2MHz}$	125 kbps BLE LR mode, Adjacent (-2 MHz) interference		-28		dB
$C/I_{125k\ BLE\ LR,+2MHz}$	125 kbps BLE LR mode, Adjacent (+2 MHz) interference		-50		dB
$C/I_{125k\ BLE\ LR,>3MHz}$	125 kbps BLE LR mode, Adjacent (≥3 MHz) interference		-55		dB
$C/I_{125k\ BLE\ LR,image}$	Image frequency interference		-29		dB

6.20.15.7 RX intermodulation

RX intermodulation[25]

Symbol	Description	Min.	Typ.	Max.	Units
$P_{IMD,5TH,1M}$	IMD performance, 1 Msps, 5th offset channel, Packet length <= 37 bytes		-33		dBm
$P_{IMD,5TH,1M,BLE}$	IMD performance, BLE 1 Msps, 5th offset channel, Packet length <= 37 bytes		-30		dBm
$P_{IMD,5TH,2M}$	IMD performance, 2 Msps, 5th offset channel, Packet length <= 37 bytes		-33		dBm
$P_{IMD,5TH,2M,BLE}$	IMD performance, BLE 2 Msps, 5th offset channel, Packet length <= 37 bytes		-31		dBm

6.20.15.8 Radio timing

Symbol	Description	Min.	Typ.	Max.	Units
t_{TXEN}	Time between TXEN task and READY event after channel FREQUENCY configured		140		µs
$t_{TXEN,FAST}$	Time between TXEN task and READY event after channel FREQUENCY configured (fast Mode)		40		µs
$t_{TXDISABLE}$	Time between DISABLE task and DISABLED event when the radio was in TX and mode is set to 1Mbps		6		µs
$t_{TXDISABLE,2M}$	Time between DISABLE task and DISABLED event when the radio was in TX and mode is set to 2Mbps		4		µs
t_{RXEN}	Time between the RXEN task and READY event after channel FREQUENCY configured in default mode		140		µs
$t_{RXEN,FAST}$	Time between the RXEN task and READY event after channel FREQUENCY configured in fast mode		40		µs

[25] Wanted signal level at PIN = -64 dBm. Two interferers with equal input power are used. The interferer closest in frequency is not modulated, the other interferer is modulated equal with the wanted signal. The input power of the interferers where the sensitivity equals BER = 0.1% is presented.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{RXDISABLE}$	Time between DISABLE task and DISABLED event when the radio was in RX		0		µs
$t_{RX\text{-}to\text{-}TX\ turnaround}$	Maximum TX-to-RX or RX-to-TX turnaround time in IEEE 802.15.4 mode	..	..	..	µs

6.20.15.9 Received signal strength indicator (RSSI) specifications

Symbol	Description	Min.	Typ.	Max.	Units
$RSSI_{ACC}$	RSSI accuracy valid range -90 to -20 dBm		+-2		dB
$RSSI_{RESOLUTION}$	RSSI resolution		1		dB
$RSSI_{PERIOD}$	Sample period		15.0		µs

6.20.15.10 Jitter

Symbol	Description	Min.	Typ.	Max.	Units
$t_{DISABLEDJITTER}$	Jitter on DISABLED event relative to END event when shortcut between END and DISABLE is enabled		0.25		µs
$t_{READYJITTER}$	Jitter on READY event relative to TXEN and RXEN task		0.25		µs

6.20.15.11 Delay when disabling the RADIO

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TXDISABLE,1M}$	Disable delay from TX Delay between DISABLE and DISABLED for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		6		µs
$t_{RXDISABLE,1M}$	Disable delay from RX. Delay between DISABLE and DISABLED for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		0		µs

6.21 RNG — Random number generator

The Random number generator (RNG) generates true non-deterministic random numbers based on internal thermal noise that are suitable for cryptographic purposes. The RNG does not require a seed value.

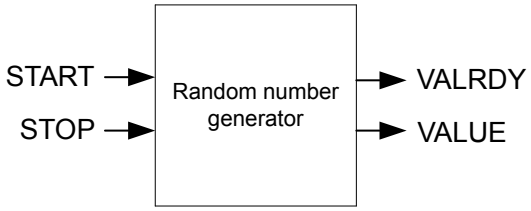

Figure 129: Random number generator

The RNG is started by triggering the START task and stopped by triggering the STOP task. When started, new random numbers are generated continuously and written to the VALUE register when ready. A VALRDY event is generated for every new random number that is written to the VALUE register. This means that after a VALRDY event is generated the CPU has the time until the next VALRDY event to read out the random number from the VALUE register before it is overwritten by a new random number.

6.21.1 Bias correction

A bias correction algorithm is employed on the internal bit stream to remove any bias toward '1' or '0'. The bits are then queued into an eight-bit register for parallel readout from the VALUE register.

It is possible to enable bias correction in the CONFIG register. This will result in slower value generation, but will ensure a statistically uniform distribution of the random values.

6.21.2 Speed

The time needed to generate one random byte of data is unpredictable, and may vary from one byte to the next. This is especially true when bias correction is enabled.

6.21.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000D000	RNG	RNG	Random number generator	

Table 87: Instances

Register	Offset	Description
TASKS_START	0x000	Task starting the random number generator
TASKS_STOP	0x004	Task stopping the random number generator
EVENTS_VALRDY	0x100	Event being generated for every new random number written to the VALUE register
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
CONFIG	0x504	Configuration register
VALUE	0x508	Output random number

Table 88: Register overview

6.21.3.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	VALRDY_STOP			Shortcut between VALRDY event and STOP task
					See EVENTS_VALRDY and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.21.3.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	VALRDY			Write '1' to enable interrupt for VALRDY event See EVENTS_VALRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.3.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	VALRDY			Write '1' to disable interrupt for VALRDY event See EVENTS_VALRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.3.4 CONFIG

Address offset: 0x504

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DERCEN			Bias correction
			Disabled	0	Disabled
			Enabled	1	Enabled

6.21.3.5 VALUE

Address offset: 0x508

Output random number

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	VALUE		[0..255]	Generated random number

6.21.4 Electrical specification

6.21.4.1 RNG Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{RNG,START}$	Time from setting the START task to generation begins. This is a one-time delay on START signal and does not apply between samples.		128		µs
$t_{RNG,RAW}$	Run time per byte without bias correction. Uniform distribution of 0 and 1 is not guaranteed.		30		µs
$t_{RNG,BC}$	Run time per byte with bias correction. Uniform distribution of 0 and 1 is guaranteed. Time to generate a byte cannot be guaranteed.		120		µs

6.22 RTC — Real-time counter

The Real-time counter (RTC) module provides a generic, low power timer on the low-frequency clock source (LFCLK).

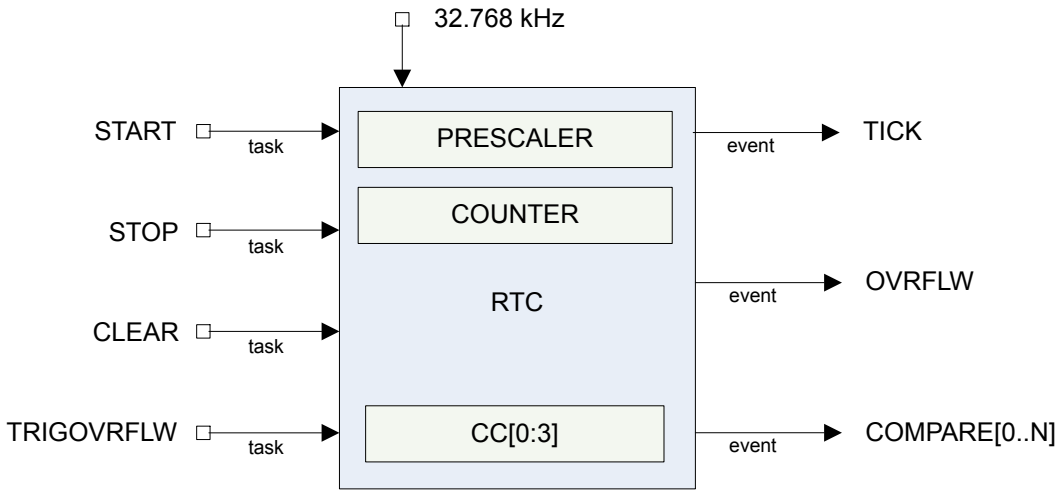

Figure 130: RTC block schematic

The RTC module features a 24-bit COUNTER, a 12-bit (1/X) prescaler, capture/compare registers, and a tick event generator for low power, tickless RTOS implementation.

6.22.1 Clock source

The RTC will run off the LFCLK.

The COUNTER resolution will therefore be 30.517 µs. Depending on the source, the RTC is able to run while the HFCLK is OFF and PCLK16M is not available.

The software has to explicitely start LFCLK before using the RTC.

See CLOCK — Clock control on page 80 for more information about clock sources.

6.22.2 Resolution versus overflow and the PRESCALER

Counter increment frequency:

```
fRTC [kHz] = 32.768 / (PRESCALER + 1 )
```

The PRESCALER register is read/write when the RTC is stopped. The PRESCALER register is read-only once the RTC is STARTed. Writing to the PRESCALER register when the RTC is started has no effect.

The PRESCALER is restarted on START, CLEAR and TRIGOVRFLW, that is, the prescaler value is latched to an internal register (<<PRESC>>) on these tasks.

Examples:

1. Desired COUNTER frequency 100 Hz (10 ms counter period)

   PRESCALER = round(32.768 kHz / 100 Hz) - 1 = 327

   f_{RTC} = 99.9 Hz

   10009.576 µs counter period
2. Desired COUNTER frequency 8 Hz (125 ms counter period)

   PRESCALER = round(32.768 kHz / 8 Hz) – 1 = 4095

   f_{RTC} = 8 Hz

   125 ms counter period

Prescaler	Counter resolution	Overflow
0	30.517 µs	512 seconds
2^{8}-1	7812.5 µs	131072 seconds
2^{12}-1	125 ms	582.542 hours

Table 89: RTC resolution versus overflow

6.22.3 COUNTER register

The COUNTER increments on LFCLK when the internal PRESCALER register (<<PRESC>>) is 0x00. <<PRESC>> is reloaded from the PRESCALER register. If enabled, the TICK event occurs on each increment of the COUNTER. The TICK event is disabled by default.

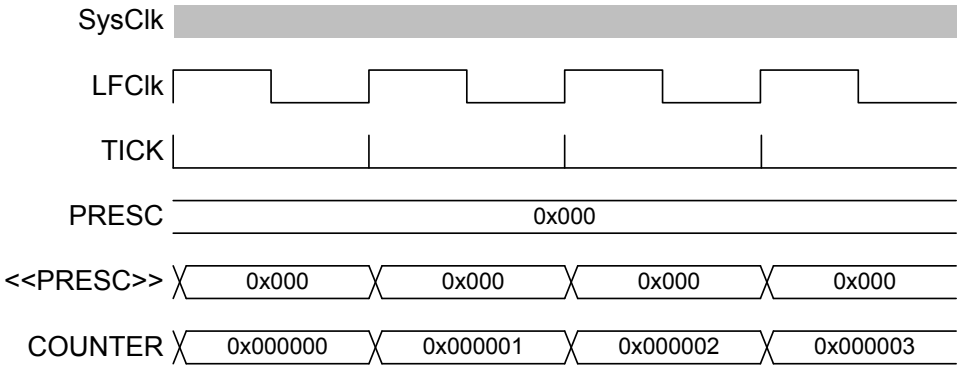

Figure 131: Timing diagram - COUNTER_PRESCALER_0

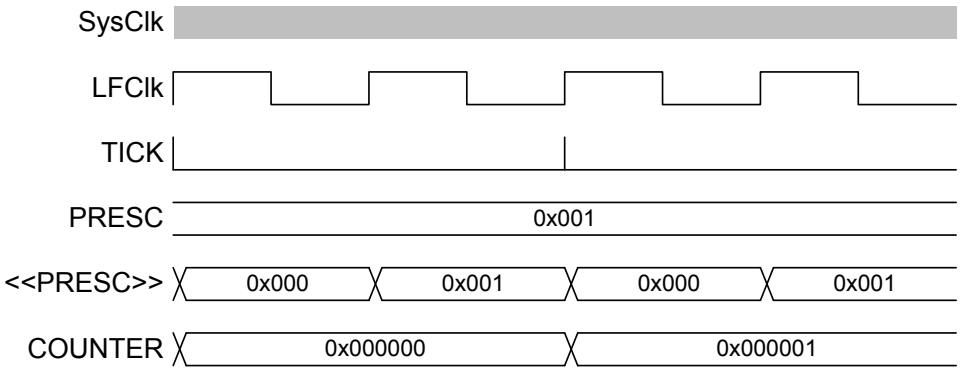

Figure 132: Timing diagram - COUNTER_PRESCALER_1

6.22.4 Overflow features

The TRIGOVRFLW task sets the COUNTER value to 0xFFFFF0 to allow SW test of the overflow condition.

OVRFLW occurs when COUNTER overflows from 0xFFFFFF to 0.

Important: The OVRFLW event is disabled by default.

6.22.5 TICK event

The TICK event enables low power "tick-less" RTOS implementation as it optionally provides a regular interrupt source for a RTOS without the need to use the ARM® SysTick feature.

Using the RTC TICK event rather than the SysTick allows the CPU to be powered down while still keeping RTOS scheduling active.

Important: The TICK event is disabled by default.

6.22.6 Event control feature

To optimize RTC power consumption, events in the RTC can be individually disabled to prevent PCLK16M and HFCLK being requested when those events are triggered. This is managed using the EVTEN register.

For example, if the TICK event is not required for an application, this event should be disabled as it is frequently occurring and may increase power consumption if HFCLK otherwise could be powered down for long durations.

This means that the RTC implements a slightly different task and event system compared to the standard system described in Peripheral interface on page 93. The RTC task and event system is illustrated in Tasks, events and interrupts in the RTC on page 336.

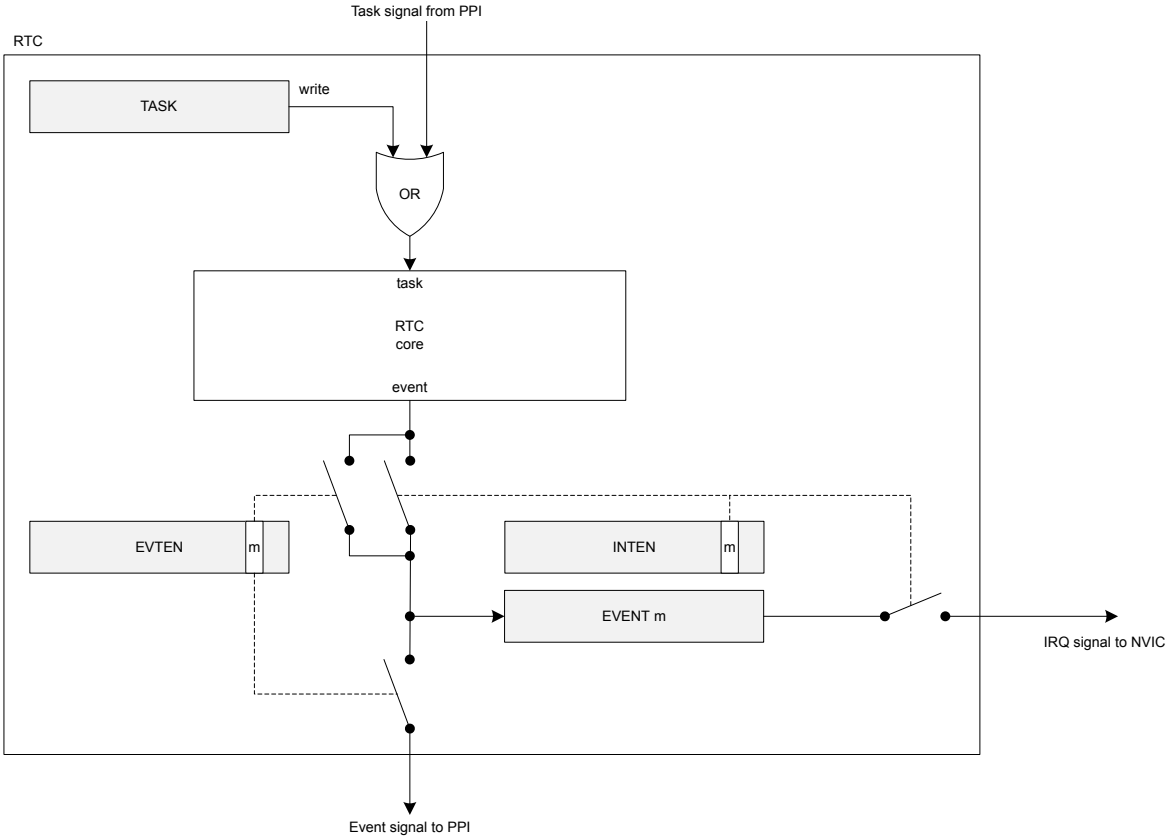

Figure 133: Tasks, events and interrupts in the RTC

6.22.7 Compare feature

There are a number of Compare registers.

For more information, see Registers on page 341.

When setting a compare register, the following behavior of the RTC compare event should be noted:

- If a CC register value is 0 when a CLEAR task is set, this will not trigger a COMPARE event.

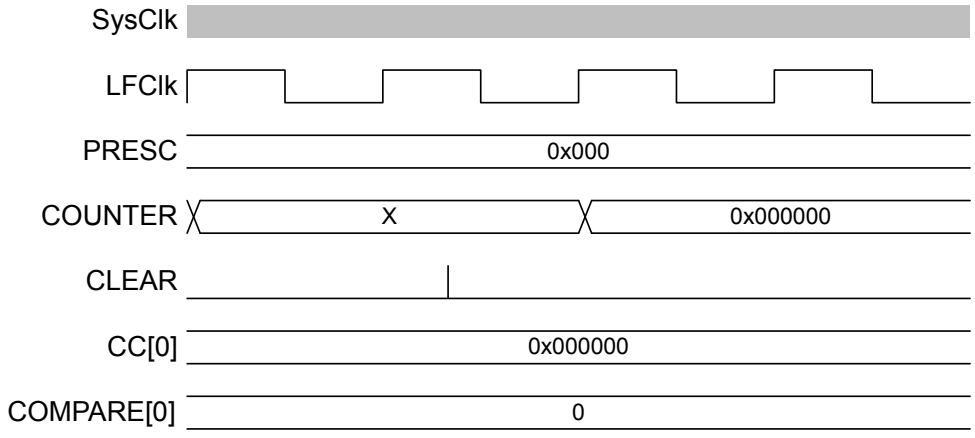

Figure 134: Timing diagram - COMPARE_CLEAR

- If a CC register is N and the COUNTER value is N when the START task is set, this will not trigger a COMPARE event.

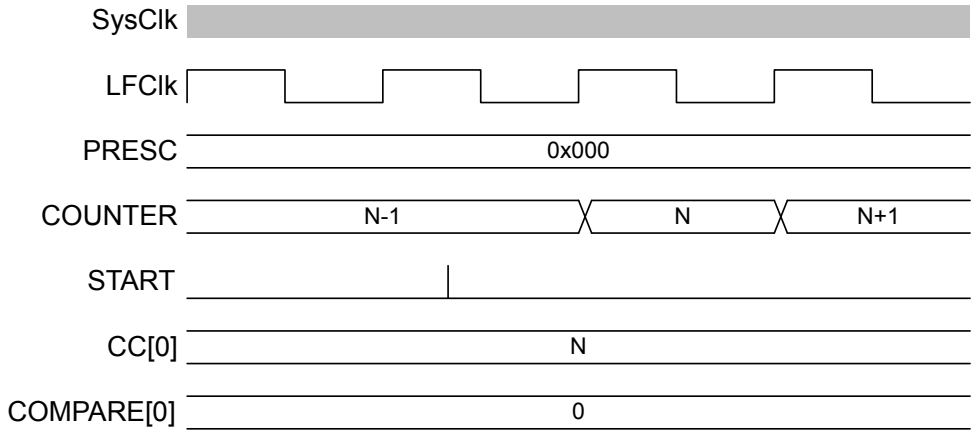

Figure 135: Timing diagram - COMPARE_START

- COMPARE occurs when a CC register is N and the COUNTER value transitions from N-1 to N.

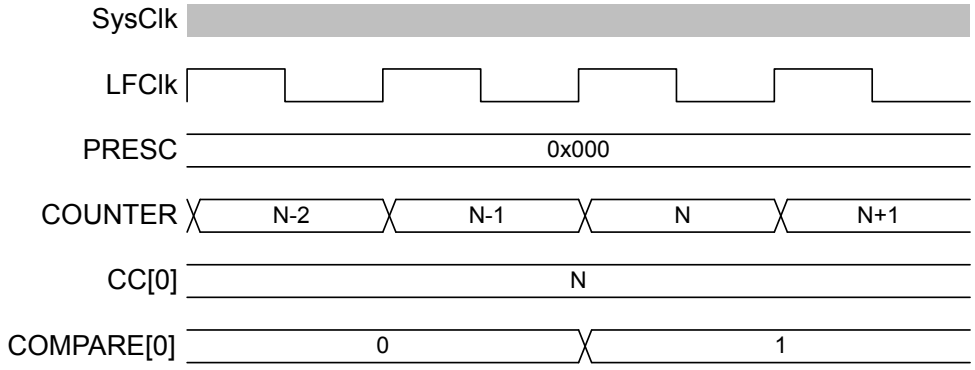

Figure 136: Timing diagram - COMPARE

- If the COUNTER is N, writing N+2 to a CC register is guaranteed to trigger a COMPARE event at N+2.

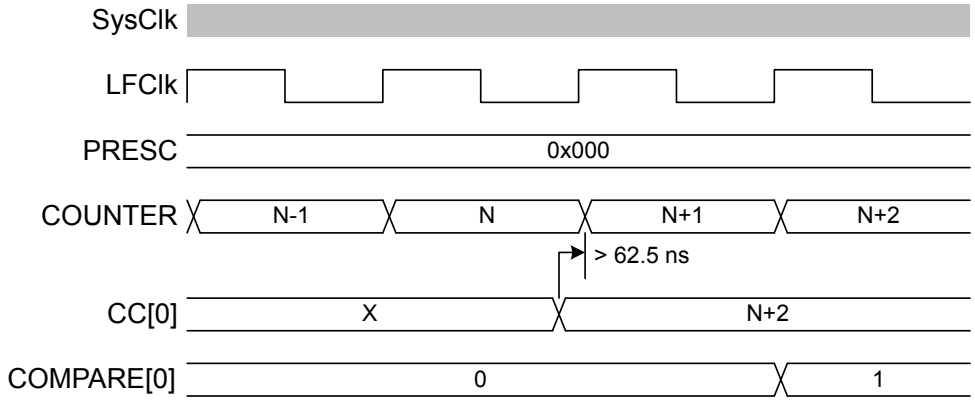

Figure 137: Timing diagram - COMPARE_N+2

- If the COUNTER is N, writing N or N+1 to a CC register may not trigger a COMPARE event.

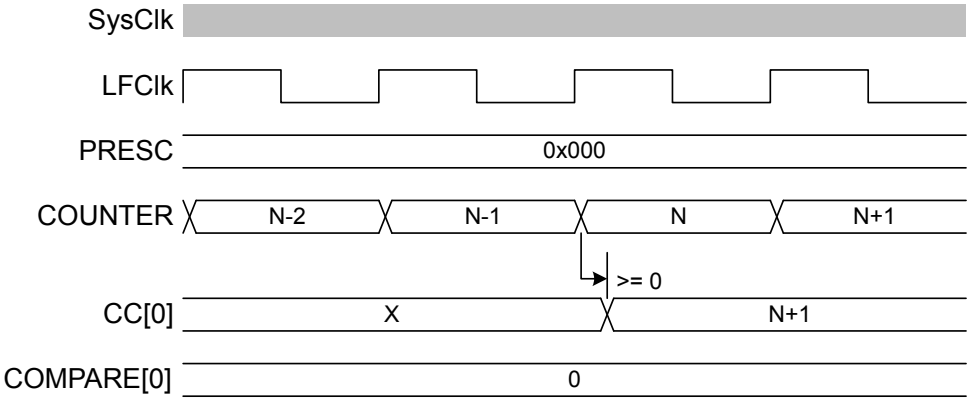

Figure 138: Timing diagram - COMPARE_N+1

- If the COUNTER is N and the current CC register value is N+1 or N+2 when a new CC value is written, a match may trigger on the previous CC value before the new value takes effect. If the current CC value greater than N+2 when the new value is written, there will be no event due to the old value.

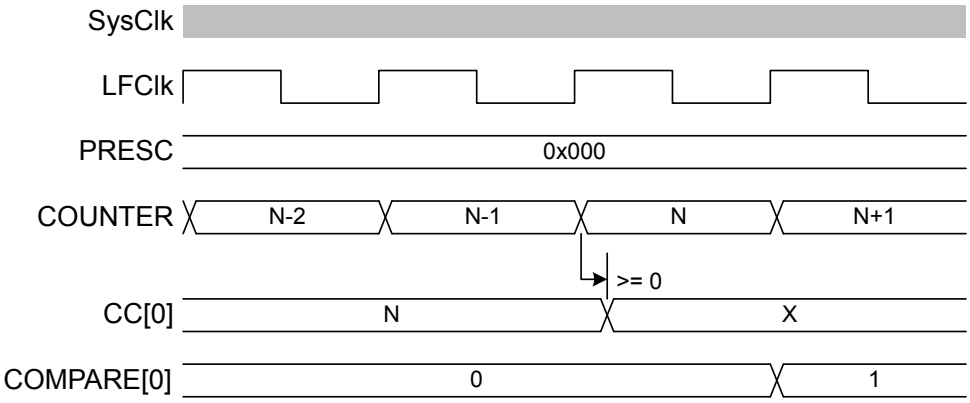

Figure 139: Timing diagram - COMPARE_N-1

6.22.8 TASK and EVENT jitter/delay

Jitter or delay in the RTC is due to the peripheral clock being a low frequency clock (LFCLK) which is not synchronous to the faster PCLK16M.

Registers in the peripheral interface, part of the PCLK16M domain, have a set of mirrored registers in the LFCLK domain. For example, the COUNTER value accessible from the CPU is in the PCLK16M domain and is latched on read from an internal register called COUNTER in the LFCLK domain. COUNTER is the register which is actually modified each time the RTC ticks. These registers must be synchronised between clock domains (PCLK16M and LFCLK).

The following is a summary of the jitter introduced on tasks and events. Figures illustrating jitter follow.

Task	Delay
CLEAR, STOP, START, TRIGOVRFLOW	+15 to 46 µs

Table 90: RTC jitter magnitudes on tasks

Operation/Function	Jitter
START to COUNTER increment	+/- 15 µs
COMPARE to COMPARE [26]	+/- 62.5 ns

Table 91: RTC jitter magnitudes on events

1. CLEAR and STOP (and TRIGOVRFLW; not shown) will be delayed as long as it takes for the peripheral to clock a falling edge and rising of the LFCLK. This is between 15.2585 µs and 45.7755 µs – rounded to 15 µs and 46 µs for the remainder of the section.

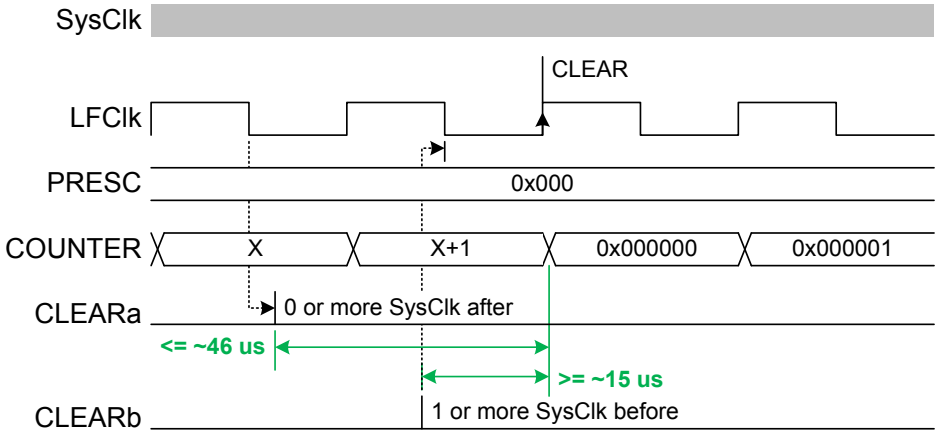

Figure 140: Timing diagram - DELAY_CLEAR

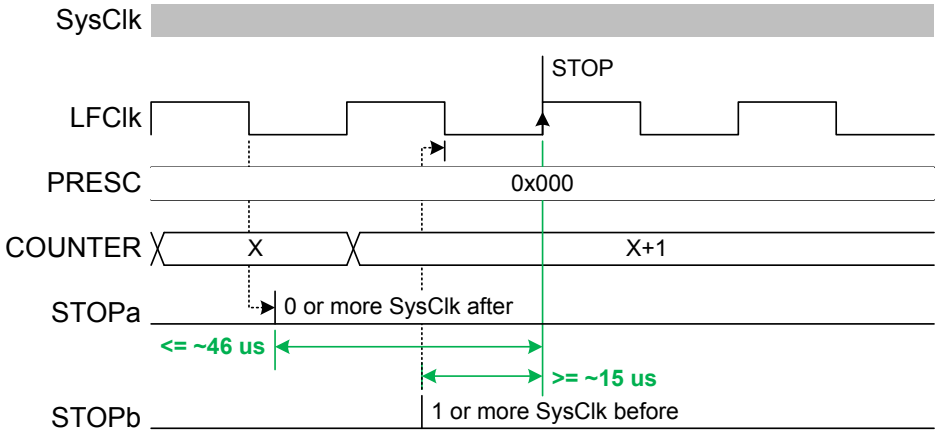

Figure 141: Timing diagram - DELAY_STOP

2. The START task will start the RTC. Assuming that the LFCLK was previously running and stable, the first increment of COUNTER (and instance of TICK event) will be typically after 30.5 µs +/-15 µs. In some cases, in particular if the RTC is STARTed before the LFCLK is running, that timing can be up to ~250 µs. The software should therefore wait for the first TICK if it has to make sure the RTC is running. Sending a TRIGOVRFLW task sets the COUNTER to a value close to overflow. However, since the update of COUNTER relies on a stable LFCLK, sending this task while LFCLK is not running will start LFCLK, but the update will then be delayed by the same amount of time of up to ~250 us. The figures show the smallest and largest delays to on the START task which appears as a +/-15 µs jitter on the first COUNTER increment.

[26] Assumes RTC runs continuously between these events.

Note: 32.768 kHz clock jitter is additional to the numbers provided above.

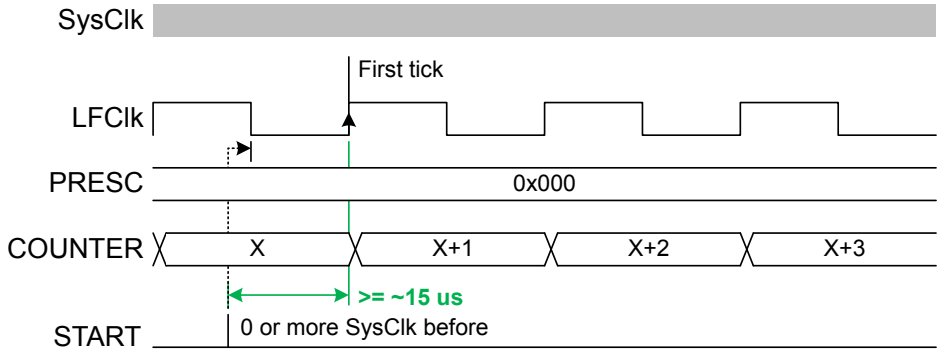

Figure 142: Timing diagram - JITTER_START-

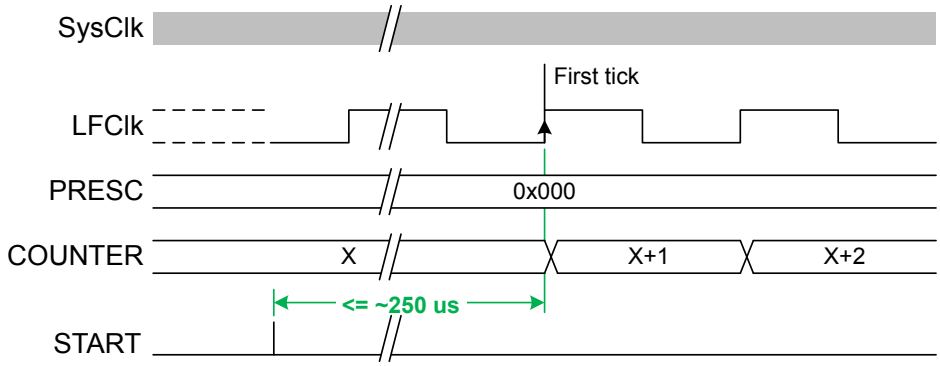

Figure 143: Timing diagram - JITTER_START+

6.22.9 Reading the COUNTER register

To read the COUNTER register, the internal <<COUNTER>> value is sampled.

To ensure that the <<COUNTER>> is safely sampled (considering an LFCLK transition may occur during a read), the CPU and core memory bus are halted for three cycles by lowering the core PREADY signal. The Read takes the CPU 2 cycles in addition resulting in the COUNTER register read taking a fixed five PCLK16M clock cycles.

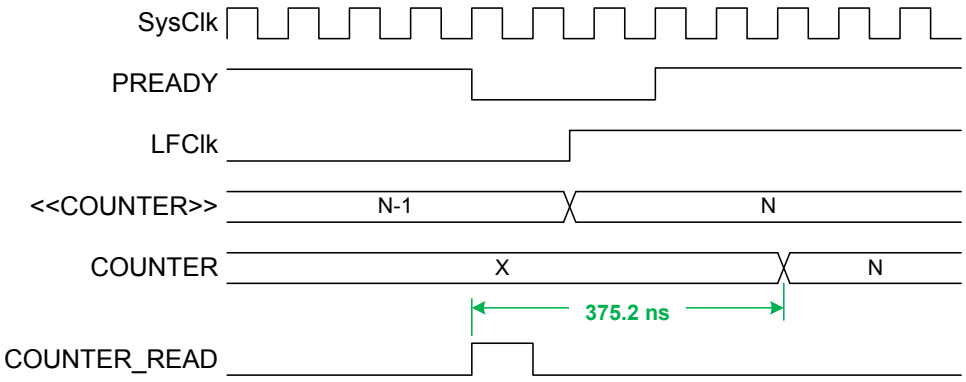

Figure 144: Timing diagram - COUNTER_READ

6.22.10 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000B000	RTC	RTC0	Real-time counter 0	CC[0..2] implemented, CC[3] not implemented
0x40011000	RTC	RTC1	Real-time counter 1	CC[0..3] implemented
0x40024000	RTC	RTC2	Real-time counter 2	CC[0..3] implemented

Table 92: Instances

Register	Offset	Description
TASKS_START	0x000	Start RTC COUNTER
TASKS_STOP	0x004	Stop RTC COUNTER
TASKS_CLEAR	0x008	Clear RTC COUNTER
TASKS_TRIGOVRFLW	0x00C	Set COUNTER to 0xFFFFF0
EVENTS_TICK	0x100	Event on COUNTER increment
EVENTS_OVRFLW	0x104	Event on COUNTER overflow
EVENTS_COMPARE[0]	0x140	Compare event on CC[0] match
EVENTS_COMPARE[1]	0x144	Compare event on CC[1] match
EVENTS_COMPARE[2]	0x148	Compare event on CC[2] match
EVENTS_COMPARE[3]	0x14C	Compare event on CC[3] match
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
EVTEN	0x340	Enable or disable event routing
EVTENSET	0x344	Enable event routing
EVTENCLR	0x348	Disable event routing
COUNTER	0x504	Current COUNTER value
PRESCALER	0x508	12 bit prescaler for COUNTER frequency (32768/(PRESCALER+1)).Must be written when RTC is stopped
CC[0]	0x540	Compare register 0
CC[1]	0x544	Compare register 1
CC[2]	0x548	Compare register 2
CC[3]	0x54C	Compare register 3

Table 93: Register overview

6.22.10.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to enable interrupt for TICK event See EVENTS_TICK
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to enable interrupt for OVRFLW event See EVENTS_OVRFLW

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE0			Write '1' to enable interrupt for COMPARE[0] event See EVENTS_COMPARE[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	COMPARE1			Write '1' to enable interrupt for COMPARE[1] event See EVENTS_COMPARE[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE2			Write '1' to enable interrupt for COMPARE[2] event See EVENTS_COMPARE[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to enable interrupt for COMPARE[3] event See EVENTS_COMPARE[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.10.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to disable interrupt for TICK event See EVENTS_TICK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to disable interrupt for OVRFLW event See EVENTS_OVRFLW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE0			Write '1' to disable interrupt for COMPARE[0] event See EVENTS_COMPARE[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
D	RW	COMPARE1			Write '1' to disable interrupt for COMPARE[1] event See EVENTS_COMPARE[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE2			Write '1' to disable interrupt for COMPARE[2] event See EVENTS_COMPARE[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to disable interrupt for COMPARE[3] event See EVENTS_COMPARE[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.10.3 EVTEN

Address offset: 0x340

Enable or disable event routing

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TICK			Enable or disable event routing for TICK event See EVENTS_TICK
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	OVRFLW			Enable or disable event routing for OVRFLW event See EVENTS_OVRFLW
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	COMPARE0			Enable or disable event routing for COMPARE[0] event See EVENTS_COMPARE[0]
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	COMPARE1			Enable or disable event routing for COMPARE[1] event See EVENTS_COMPARE[1]
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	COMPARE2			Enable or disable event routing for COMPARE[2] event See EVENTS_COMPARE[2]
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
F	RW	COMPARE3			Enable or disable event routing for COMPARE[3] event See EVENTS_COMPARE[3]
			Disabled	0	Disable
			Enabled	1	Enable

6.22.10.4 EVTENSET

Address offset: 0x344

Enable event routing

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to enable event routing for TICK event See EVENTS_TICK
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to enable event routing for OVRFLW event See EVENTS_OVRFLW
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE0			Write '1' to enable event routing for COMPARE[0] event See EVENTS_COMPARE[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	COMPARE1			Write '1' to enable event routing for COMPARE[1] event See EVENTS_COMPARE[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE2			Write '1' to enable event routing for COMPARE[2] event See EVENTS_COMPARE[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to enable event routing for COMPARE[3] event See EVENTS_COMPARE[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.10.5 EVTENCLR

Address offset: 0x348

Disable event routing

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to disable event routing for TICK event See EVENTS_TICK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to disable event routing for OVRFLW event See EVENTS_OVRFLW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE0			Write '1' to disable event routing for COMPARE[0] event See EVENTS_COMPARE[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	COMPARE1			Write '1' to disable event routing for COMPARE[1] event See EVENTS_COMPARE[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE2			Write '1' to disable event routing for COMPARE[2] event See EVENTS_COMPARE[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to disable event routing for COMPARE[3] event See EVENTS_COMPARE[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.10.6 COUNTER

Address offset: 0x504

Current COUNTER value

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	COUNTER			Counter value

6.22.10.7 PRESCALER

Address offset: 0x508

12 bit prescaler for COUNTER frequency (32768/(PRESCALER+1)).Must be written when RTC is stopped

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PRESCALER			Prescaler value

6.22.10.8 CC[n] (n=0..3)

Address offset: 0x540 + (n × 0x4)

Compare register n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	COMPARE			Compare value

6.22.11 Electrical specification

6.23 SAADC — Successive approximation analog-to-digital converter

The SAADC is a differential successive approximation register (SAR) analog-to-digital converter. It supports up to eight external analog input channels, depending on package variant.

The following lists the main features of the SAADC:

- Multiple input channels
  - Each channel can use pins AIN0 through AIN7, the VDD pin, or the VDDH pin as input
  - Eight channels for single-ended inputs and four channels for differential inputs
- Full scale input range
- Individual reference selection for each channel
  - VDD
  - Internal reference
- Continuous sampling
- Output samples are automatically written to RAM using EasyDMA
- Samples are stored as 16-bit 2's complement values
- 8/10/12-bit resolution, 14-bit resolution with oversampling

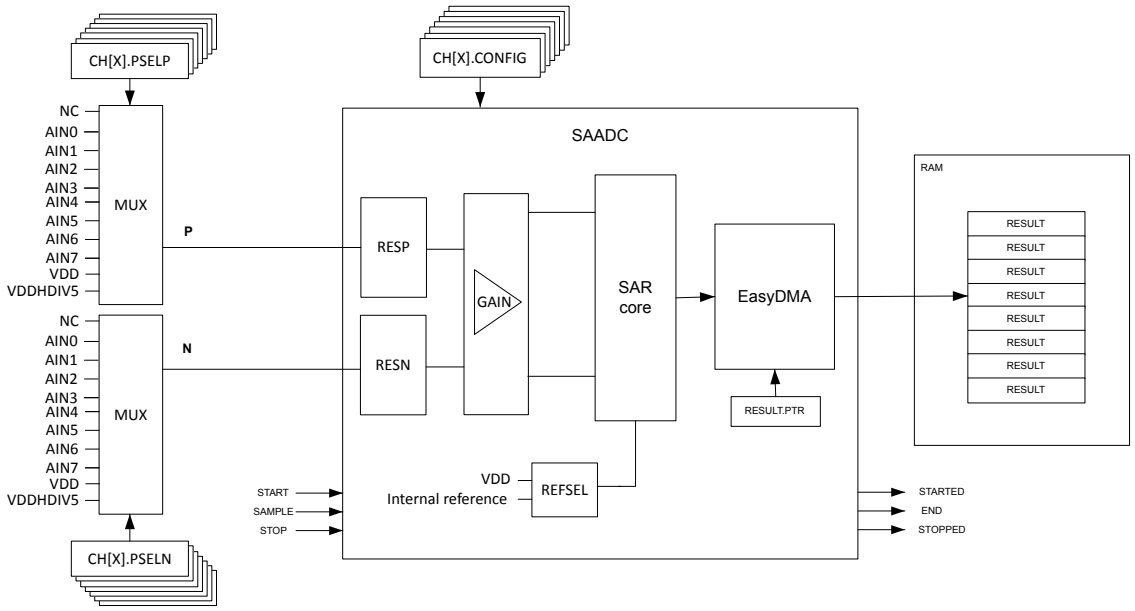

Figure 145: Block diagram

An input channel is enabled and connected to an analog input pin using the registers CH[n].PSELP (n=0..7) on page 362 and CH[n].PSELN (n=0..7) on page 362.

Before any sampling can take place, the length and the location of the memory buffer in RAM where output values shall be written needs to be configured, and the START task has to be triggered to apply the configuration. See EasyDMA on page 349 for details on memory configuration and how the results are placed in memory.

Sampling of a single channel is started by triggering the SAMPLE task, and the sample results are automatically written to memory using EasyDMA. A DONE event is generated for every single completed conversion, and an END event is generated when multiple samples, as specified in RESULT.MAXCNT on page 365, have been written to memory.

When multiple channels are enabled, they are sampled successively in a sequence and in circular order (lowest to highest channel number, and then from lowest again). The SAMPLE task needs to be triggered for every individual sample to be taken, for every channel.

6.23.1 Input configuration

Each SAADC channel can be configured to use either single-ended or differential input mode.

The configuration is done using the registers CH[n].CONFIG (n=0..7) on page 362. In single-ended mode, the negative channel input is shorted to ground internally and the setting in the corresponding register CH[n].PSELN (n=0..7) on page 362 will not apply. The assumption in single-ended mode is that the internal ground of the SAADC is the same as the external ground that the measured voltage is referred to. The SAADC is thus sensitive to ground bounce on the PCB in single-ended mode. If this is a concern, using differential measurement is recommended. In differential mode, both positive and negative input has to be configured in registersCH[n].PSELP (n=0..7) on page 362 and CH[n].PSELN (n=0..7) on page 362 respectively.

6.23.1.1 Acquisition time

To sample input voltage, the SAADC connects a capacitor to the input.

This is illustrated in the following figure:

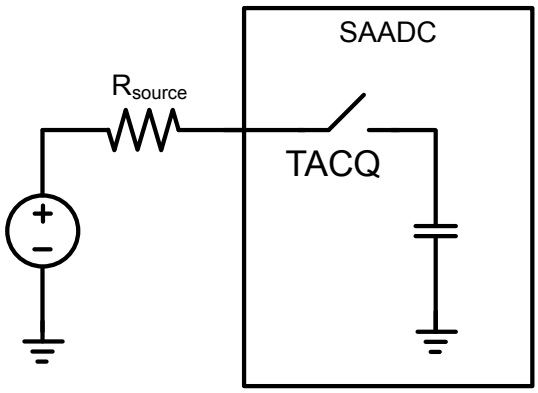

Figure 146: Simplified SAADC sample network

The acquisition time indicates how long the capacitor is connected, see TACQ field in CH[n].CONFIG register. The required acquisition time depends on the source resistance (R_{source}). For high source resistance the acquisition time should be increased:

TACQ [µs]	Maximum source resistance [kΩ]
3	10
5	40
10	100
15	200
20	400
40	800

Table 94: Acquisition time

When using `VDDHDIV5` as input, the acquisition time needs to be 10 µs or higher.

6.23.1.2 Internal resistor string (resistor ladder)

The SAADC has an internal resistor string for positive and negative input. The resistors are controlled in registers CH[n].CONFIG.RESP and CH[n].CONFIG.RESN.

The following figure illustrates the resistor ladder for positive (and negative) input:

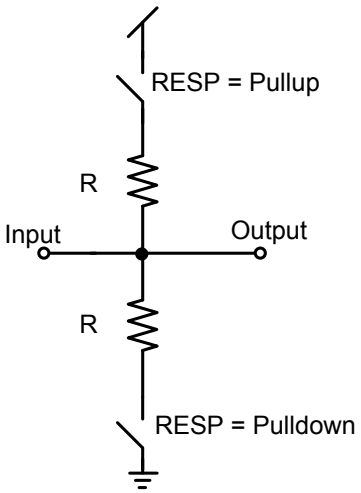

Figure 147: Resistor ladder for positive input (negative input is equivalent, using RESN instead of RESP)

6.23.2 Reference voltage and gain settings

Each SAADC channel can have individual reference and gain settings.

This is configured in registers CH[n].CONFIG (n=0..7) on page 362. Available configuration options are:

- VDD/4 or internal 0.6 V reference
- Gain ranging from 1/6 to 4

The gain setting can be used to control the effective input range of the SAADC:

```
Input range = (±0.6 V or ±VDD/4)/gain
```

For example, selecting VDD as reference, single-ended input (grounded negative input), and a gain of 1/4 will result in the following input range:

```
Input range = (VDD/4)/(1/4) = VDD
```

With internal reference, single-ended input (grounded negative input) and a gain of 1/6, the input range will be:

```
Input range = (0.6 V)/(1/6) = 3.6 V
```

Inputs AIN0 through AIN7 cannot exceed VDD or be lower than VSS.

6.23.3 Digital output

The digital output value from the SAADC is calculated using a formula.

$$RESULT = (V(P) - V(N)) * (GAIN/REFERENCE) * 2^{(RESOLUTION - m)}$$

where

V(P)
: is the voltage at input P

V(N)
: is the voltage at input N

GAIN
: is the selected gain

REFERENCE
: is the selected reference voltage

RESOLUTION
: is output resolution in bits, as configured in register RESOLUTION on page 364

m
: is 0 for single-ended channels

: is 1 for differential channels

Results are sign extended to 16 bits and stored as little-endian byte order in RAM.

6.23.4 EasyDMA

The SAADC resources are started by triggering the START task. The SAADC is using EasyDMA to store results in a result buffer in RAM.

Registers RESULT.PTR on page 365 and RESULT.MAXCNT on page 365 must be configured before SAADC is started.

The result buffer is located at the address specified in register RESULT.PTR on page 365. This register is double-buffered, and it can be updated and prepared for the next START task immediately after the STARTED event is generated. The size of the result buffer is specified in register RESULT.MAXCNT on page 365, and the SAADC will generate an END event when it has filled up the result buffer, as illustrated in the following figure:

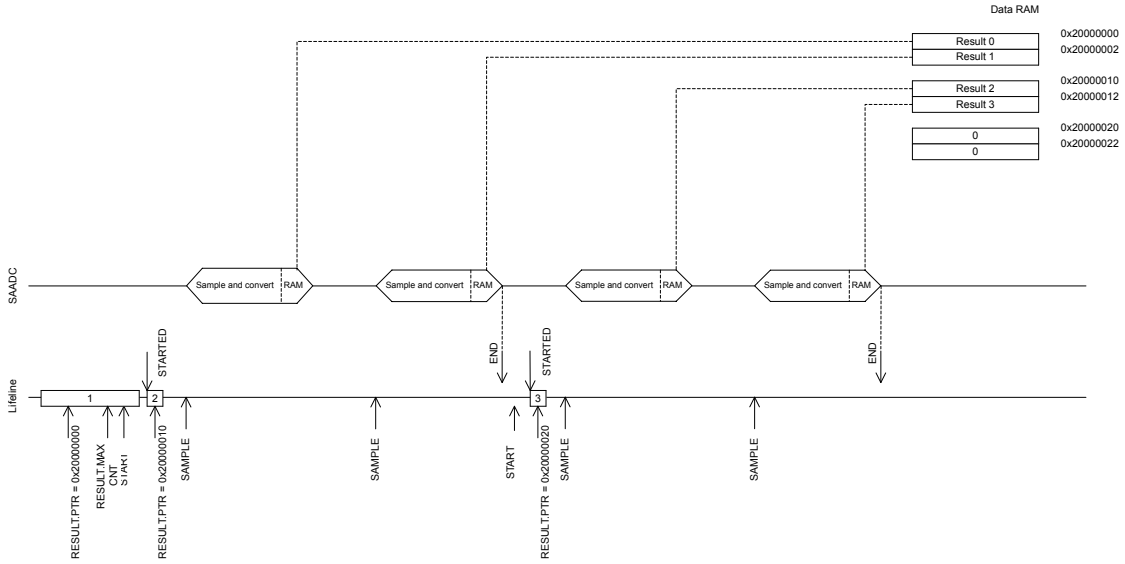

Figure 148: SAADC

The following figure shows how results are placed in RAM when multiple channels are enabled, and value in RESULT.MAXCNT on page 365 is an even number:

	31 … 16	15 … 0
RESULT.PTR	CH[2] 1st result	CH[1] 1st result
RESULT.PTR + 4	CH[1] 2nd result	CH[5] 1st result
RESULT.PTR + 8	CH[5] 2nd result	CH[2] 2nd result
	(…)	
RESULT.PTR + 2*RESULT.MAXCNT – 4	CH[5] last result	CH[2] last result

Figure 149: Example of RAM placement: RESULT.MAXCNT even number, channels 1, 2 and 5 enabled

The following figure shows how results are placed in RAM when multiple channels are enabled and value in RESULT.MAXCNT on page 365 is an odd number:

	31 … 16	15 … 0
RESULT.PTR	CH[2] 1st result	CH[1] 1st result
RESULT.PTR + 4	CH[1] 2nd result	CH[5] 1st result
RESULT.PTR + 8	CH[5] 2nd result	CH[2] 2nd result
	(…)	
RESULT.PTR + 2*RESULT.MAXCNT – 2		CH[5] last result

Figure 150: Example of RAM placement: RESULT.MAXCNT odd number, channels 1, 2 and 5 enabled

The last 32-bit word is populated only with one 16-bit result. In both examples, channels 1, 2 and 5 are enabled, and all others are disabled.

See Memory on page 19 for more information about the different memory regions.

EasyDMA is finished with accessing RAM when events END or STOPPED are generated. The register RESULT.AMOUNT on page 365 can then be read, to see how many results have been transferred to the result buffer in RAM since the START task was triggered.

6.23.5 Continuous sampling

When using continuous sampling, new samples are automatically taken at a fixed sample rate.

Continuous sampling of both single and multiple channels can be implemented using a general purpose timer connecting a timer event to SAADC's SAMPLE task via PPI.

Alternatively, continuous sampling can be implemented by using the internal timer in the SAADC by setting the MODE field in register SAMPLERATE on page 364 to `Timers`. The sample rate (frequency at which the SAMPLE task is triggered) is configured in the same register. The internal timer and the continuous sampling are started by triggering the START task and stopped using the STOP task.

> **Note:** Note that the internal timer can only be used when a single input channel is enabled.

For continuous sampling, ensure that the sample rate fullfills the following criteria:

$f_{SAMPLE} < 1/[t_{ACQ} + t_{conv}]$

6.23.6 Oversampling

An accumulator in the SAADC can be used to find the average of several analog input samples. In general, oversampling improves the signal-to-noise ratio (SNR). Oversampling does not improve the integral non-linearity (INL) or differential non-linearity (DNL).

The accumulator is controlled in the OVERSAMPLE register. When using oversampling, $2^{OVERSAMPLE}$ input samples are averaged before the sample result is transferred to memory. Hence, the SAMPLE task must be triggered $2^{OVERSAMPLE}$ times for each output value. The following events are relevant:

- DONE event is generated for every input sample taken
- RESULTDONE event is generated for every averaged value ready to be transferred into RAM
- END event is generated when averaged values defined in RESULT.MAXCNT on page 365 have been written to memory. END event is generated every $2^{OVERSAMPLE}$ time the DONE event is generated.

If value in OVERSAMPLE is set to 0, the DONE and RESULTDONE events will be generated at the same rate.

> **Note:** Oversampling should only be used when a single input channel is enabled, as averaging is performed over all enabled channels.

6.23.7 Event monitoring using limits

A channel can be event monitored by using limits.

Limits are configured in CH[n].LIMIT register, with high limit and low limit.

> **Note:** High limit shall always be higher than or equal to low limit.

Appropriate events are generated whenever the conversion results (sampled input signals) are outside of the two defined limits. It is not possible to generate an event when the input signal is inside a defined range by swapping high and low limits. An example of event montitoring using limits is illustrated in the following figure:

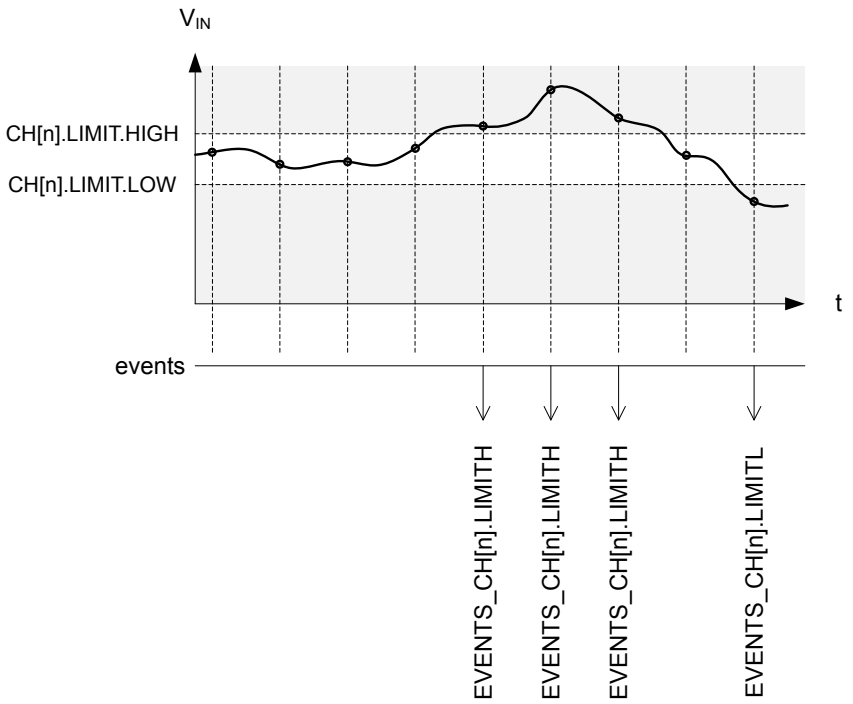

Figure 151: Example: Event monitoring on channel n using limits

The comparison to limits always takes place, it does not need to be specifically enabled. If comparison is not required on a channel, the software ignores the related events. In that situation, the value of the limits defined in register is irrelevant, i.e. it does not matter if the low limit is lower than the high limit or not.

6.23.8 Calibration

The SAADC has a temperature dependent offset.

Therefore, it is recommended to calibrate the SAADC at least once before use, and to re-run calibration every time the ambient temperature has changed by more than 10 °C.

Offset calibration is started by triggering the CALIBRATEOFFSET task, and the CALIBRATEDONE event is generated when calibration is done.

6.23.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40007000	SAADC	SAADC	Analog to digital converter	

Table 95: Instances

Register	Offset	Description
TASKS_START	0x000	Starts the SAADC and prepares the result buffer in RAM
TASKS_SAMPLE	0x004	Takes one SAADC sample
TASKS_STOP	0x008	Stops the SAADC and terminates all on-going conversions
TASKS_CALIBRATEOFFSET	0x00C	Starts offset auto-calibration
EVENTS_STARTED	0x100	The SAADC has started
EVENTS_END	0x104	The SAADC has filled up the result buffer
EVENTS_DONE	0x108	A conversion task has been completed. Depending on the configuration, multiple conversions might be needed for a result to be transferred to RAM.
EVENTS_RESULTDONE	0x10C	Result ready for transfer to RAM

Register	Offset	Description
EVENTS_CALIBRATEDONE	0x110	Calibration is complete
EVENTS_STOPPED	0x114	The SAADC has stopped
EVENTS_CH[0].LIMITH	0x118	Last result is equal or above CH[0].LIMIT.HIGH
EVENTS_CH[0].LIMITL	0x11C	Last result is equal or below CH[0].LIMIT.LOW
EVENTS_CH[1].LIMITH	0x120	Last result is equal or above CH[1].LIMIT.HIGH
EVENTS_CH[1].LIMITL	0x124	Last result is equal or below CH[1].LIMIT.LOW
EVENTS_CH[2].LIMITH	0x128	Last result is equal or above CH[2].LIMIT.HIGH
EVENTS_CH[2].LIMITL	0x12C	Last result is equal or below CH[2].LIMIT.LOW
EVENTS_CH[3].LIMITH	0x130	Last result is equal or above CH[3].LIMIT.HIGH
EVENTS_CH[3].LIMITL	0x134	Last result is equal or below CH[3].LIMIT.LOW
EVENTS_CH[4].LIMITH	0x138	Last result is equal or above CH[4].LIMIT.HIGH
EVENTS_CH[4].LIMITL	0x13C	Last result is equal or below CH[4].LIMIT.LOW
EVENTS_CH[5].LIMITH	0x140	Last result is equal or above CH[5].LIMIT.HIGH
EVENTS_CH[5].LIMITL	0x144	Last result is equal or below CH[5].LIMIT.LOW
EVENTS_CH[6].LIMITH	0x148	Last result is equal or above CH[6].LIMIT.HIGH
EVENTS_CH[6].LIMITL	0x14C	Last result is equal or below CH[6].LIMIT.LOW
EVENTS_CH[7].LIMITH	0x150	Last result is equal or above CH[7].LIMIT.HIGH
EVENTS_CH[7].LIMITL	0x154	Last result is equal or below CH[7].LIMIT.LOW
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
STATUS	0x400	Status
ENABLE	0x500	Enable or disable SAADC
CH[0].PSELP	0x510	Input positive pin selection for CH[0]
CH[0].PSELN	0x514	Input negative pin selection for CH[0]
CH[0].CONFIG	0x518	Input configuration for CH[0]
CH[0].LIMIT	0x51C	High/low limits for event monitoring of a channel
CH[1].PSELP	0x520	Input positive pin selection for CH[1]
CH[1].PSELN	0x524	Input negative pin selection for CH[1]
CH[1].CONFIG	0x528	Input configuration for CH[1]
CH[1].LIMIT	0x52C	High/low limits for event monitoring of a channel
CH[2].PSELP	0x530	Input positive pin selection for CH[2]
CH[2].PSELN	0x534	Input negative pin selection for CH[2]
CH[2].CONFIG	0x538	Input configuration for CH[2]
CH[2].LIMIT	0x53C	High/low limits for event monitoring of a channel
CH[3].PSELP	0x540	Input positive pin selection for CH[3]
CH[3].PSELN	0x544	Input negative pin selection for CH[3]
CH[3].CONFIG	0x548	Input configuration for CH[3]
CH[3].LIMIT	0x54C	High/low limits for event monitoring of a channel
CH[4].PSELP	0x550	Input positive pin selection for CH[4]
CH[4].PSELN	0x554	Input negative pin selection for CH[4]
CH[4].CONFIG	0x558	Input configuration for CH[4]
CH[4].LIMIT	0x55C	High/low limits for event monitoring of a channel
CH[5].PSELP	0x560	Input positive pin selection for CH[5]
CH[5].PSELN	0x564	Input negative pin selection for CH[5]
CH[5].CONFIG	0x568	Input configuration for CH[5]
CH[5].LIMIT	0x56C	High/low limits for event monitoring of a channel
CH[6].PSELP	0x570	Input positive pin selection for CH[6]
CH[6].PSELN	0x574	Input negative pin selection for CH[6]
CH[6].CONFIG	0x578	Input configuration for CH[6]
CH[6].LIMIT	0x57C	High/low limits for event monitoring of a channel
CH[7].PSELP	0x580	Input positive pin selection for CH[7]
CH[7].PSELN	0x584	Input negative pin selection for CH[7]

Register	Offset	Description
CH[7].CONFIG	0x588	Input configuration for CH[7]
CH[7].LIMIT	0x58C	High/low limits for event monitoring of a channel
RESOLUTION	0x5F0	Resolution configuration
OVERSAMPLE	0x5F4	Oversampling configuration. The RESOLUTION is applied before averaging, thus for high OVERSAMPLE a higher RESOLUTION should be used.
SAMPLERATE	0x5F8	Controls normal or continuous sample rate
RESULT.PTR	0x62C	Data pointer
RESULT.MAXCNT	0x630	Maximum number of 16-bit samples to be written to output RAM buffer
RESULT.AMOUNT	0x634	Number of 16-bit samples written to output RAM buffer since the previous START task

Table 96: Register overview

6.23.9.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Enable or disable interrupt for STARTED event See EVENTS_STARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	END			Enable or disable interrupt for END event See EVENTS_END
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	DONE			Enable or disable interrupt for DONE event See EVENTS_DONE
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	RESULTDONE			Enable or disable interrupt for RESULTDONE event See EVENTS_RESULTDONE
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	CALIBRATEDONE			Enable or disable interrupt for CALIBRATEDONE event See EVENTS_CALIBRATEDONE
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	CH0LIMITH			Enable or disable interrupt for CH[0].LIMITH event See EVENTS_CH[0].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
H	RW	CH0LIMITL			Enable or disable interrupt for CH[0].LIMITL event See EVENTS_CH[0].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	CH1LIMITH			Enable or disable interrupt for CH[1].LIMITH event See EVENTS_CH[1].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	CH1LIMITL			Enable or disable interrupt for CH[1].LIMITL event See EVENTS_CH[1].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	CH2LIMITH			Enable or disable interrupt for CH[2].LIMITH event See EVENTS_CH[2].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	CH2LIMITL			Enable or disable interrupt for CH[2].LIMITL event See EVENTS_CH[2].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	CH3LIMITH			Enable or disable interrupt for CH[3].LIMITH event See EVENTS_CH[3].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	CH3LIMITL			Enable or disable interrupt for CH[3].LIMITL event See EVENTS_CH[3].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	CH4LIMITH			Enable or disable interrupt for CH[4].LIMITH event See EVENTS_CH[4].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	CH4LIMITL			Enable or disable interrupt for CH[4].LIMITL event See EVENTS_CH[4].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
Q	RW	CH5LIMITH			Enable or disable interrupt for CH[5].LIMITH event See EVENTS_CH[5].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	CH5LIMITL			Enable or disable interrupt for CH[5].LIMITL event See EVENTS_CH[5].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	CH6LIMITH			Enable or disable interrupt for CH[6].LIMITH event See EVENTS_CH[6].LIMITH

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
T	RW	CH6LIMITL			Enable or disable interrupt for CH[6].LIMITL event See EVENTS_CH[6].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
U	RW	CH7LIMITH			Enable or disable interrupt for CH[7].LIMITH event See EVENTS_CH[7].LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
V	RW	CH7LIMITL			Enable or disable interrupt for CH[7].LIMITL event See EVENTS_CH[7].LIMITL
			Disabled	0	Disable
			Enabled	1	Enable

6.23.9.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to enable interrupt for STARTED event See EVENTS_STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to enable interrupt for DONE event See EVENTS_DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to enable interrupt for RESULTDONE event See EVENTS_RESULTDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to enable interrupt for CALIBRATEDONE event See EVENTS_CALIBRATEDONE
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to enable interrupt for CH[0].LIMITH event See EVENTS_CH[0].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to enable interrupt for CH[0].LIMITL event See EVENTS_CH[0].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to enable interrupt for CH[1].LIMITH event See EVENTS_CH[1].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to enable interrupt for CH[1].LIMITL event See EVENTS_CH[1].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to enable interrupt for CH[2].LIMITH event See EVENTS_CH[2].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to enable interrupt for CH[2].LIMITL event See EVENTS_CH[2].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to enable interrupt for CH[3].LIMITH event See EVENTS_CH[3].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to enable interrupt for CH[3].LIMITL event See EVENTS_CH[3].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
O	RW	CH4LIMITH			Write '1' to enable interrupt for CH[4].LIMITH event See EVENTS_CH[4].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to enable interrupt for CH[4].LIMITL event See EVENTS_CH[4].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to enable interrupt for CH[5].LIMITH event See EVENTS_CH[5].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to enable interrupt for CH[5].LIMITL event See EVENTS_CH[5].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to enable interrupt for CH[6].LIMITH event See EVENTS_CH[6].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to enable interrupt for CH[6].LIMITL event See EVENTS_CH[6].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	CH7LIMITH			Write '1' to enable interrupt for CH[7].LIMITH event See EVENTS_CH[7].LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to enable interrupt for CH[7].LIMITL event See EVENTS_CH[7].LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.23.9.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to disable interrupt for STARTED event See EVENTS_STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to disable interrupt for END event See EVENTS_END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to disable interrupt for DONE event See EVENTS_DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to disable interrupt for RESULTDONE event See EVENTS_RESULTDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to disable interrupt for CALIBRATEDONE event See EVENTS_CALIBRATEDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to disable interrupt for CH[0].LIMITH event See EVENTS_CH[0].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to disable interrupt for CH[0].LIMITL event See EVENTS_CH[0].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to disable interrupt for CH[1].LIMITH event See EVENTS_CH[1].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to disable interrupt for CH[1].LIMITL event See EVENTS_CH[1].LIMITL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to disable interrupt for CH[2].LIMITH event See EVENTS_CH[2].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to disable interrupt for CH[2].LIMITL event See EVENTS_CH[2].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to disable interrupt for CH[3].LIMITH event See EVENTS_CH[3].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to disable interrupt for CH[3].LIMITL event See EVENTS_CH[3].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	CH4LIMITH			Write '1' to disable interrupt for CH[4].LIMITH event See EVENTS_CH[4].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to disable interrupt for CH[4].LIMITL event See EVENTS_CH[4].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to disable interrupt for CH[5].LIMITH event See EVENTS_CH[5].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to disable interrupt for CH[5].LIMITL event See EVENTS_CH[5].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to disable interrupt for CH[6].LIMITH event See EVENTS_CH[6].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to disable interrupt for CH[6].LIMITL event See EVENTS_CH[6].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	CH7LIMITH			Write '1' to disable interrupt for CH[7].LIMITH event See EVENTS_CH[7].LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to disable interrupt for CH[7].LIMITL event See EVENTS_CH[7].LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.23.9.4 STATUS

Address offset: 0x400

Status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	STATUS			Status
			Ready	0	SAADC is ready. No on-going conversions.
			Busy	1	SAADC is busy. Conversion in progress.

6.23.9.5 ENABLE

Address offset: 0x500

Enable or disable SAADC

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SAADC
			Disabled	0	Disable SAADC
			Enabled	1	Enable SAADC When enabled, the SAADC will acquire access to analog input pins specified in registers CH[n].PSELP and CH[n].PSELN

6.23.9.6 CH[n].PSELP (n=0..7)

Address offset: 0x510 + (n × 0x10)

Input positive pin selection for CH[n]

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELP			Analog positive input channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD
			VDDHDIV5	0x0D	VDDH/5

6.23.9.7 CH[n].PSELN (n=0..7)

Address offset: 0x514 + (n × 0x10)

Input negative pin selection for CH[n]

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELN			Analog negative input, enables differential channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD
			VDDHDIV5	0x0D	VDDH/5

6.23.9.8 CH[n].CONFIG (n=0..7)

Address offset: 0x518 + (n × 0x10)

Input configuration for CH[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E E E D C C C B B A A
Reset 0x00020000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RESP			Positive channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
B	RW	RESN			Negative channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
C	RW	GAIN			Gain control
			Gain1_6	0	1/6
			Gain1_5	1	1/5
			Gain1_4	2	1/4
			Gain1_3	3	1/3
			Gain1_2	4	1/2
			Gain1	5	1
			Gain2	6	2
			Gain4	7	4
D	RW	REFSEL			Reference control
			Internal	0	Internal reference (0.6 V)
			VDD1_4	1	VDD/4 as reference
E	RW	TACQ			Acquisition time, the time the SAADC uses to sample the input voltage
			3us	0	3 µs
			5us	1	5 µs
			10us	2	10 µs
			15us	3	15 µs
			20us	4	20 µs
			40us	5	40 µs
F	RW	MODE			Enable differential mode
			SE	0	Single-ended, PSELN will be ignored, negative input to SAADC shorted to GND
			Diff	1	Differential
G	RW	BURST			Enable burst mode
			Disabled	0	Burst mode is disabled (normal operation)
			Enabled	1	Burst mode is enabled. SAADC takes 2^OVERSAMPLE number of samples as fast as it can, and sends the average to Data RAM.

6.23.9.9 CH[n].LIMIT (n=0..7)

Address offset: 0x51C + (n × 0x10)

High/low limits for event monitoring of a channel

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B B B B B B B B B A A A A A A A A A A A A A A
Reset 0x7FFF8000				**0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LOW		[-32768 to +32767]	Low level limit
B	RW	HIGH		[-32768 to +32767]	High level limit

6.23.9.10 RESOLUTION

Address offset: 0x5F0

Resolution configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	VAL			Set the resolution
			8bit	0	8 bits
			10bit	1	10 bits
			12bit	2	12 bits
			14bit	3	14 bits

6.23.9.11 OVERSAMPLE

Address offset: 0x5F4

Oversampling configuration. The RESOLUTION is applied before averaging, thus for high OVERSAMPLE a higher RESOLUTION should be used.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERSAMPLE			Oversample control
			Bypass	0	Bypass oversampling
			Over2x	1	Oversample 2x
			Over4x	2	Oversample 4x
			Over8x	3	Oversample 8x
			Over16x	4	Oversample 16x
			Over32x	5	Oversample 32x
			Over64x	6	Oversample 64x
			Over128x	7	Oversample 128x
			Over256x	8	Oversample 256x

6.23.9.12 SAMPLERATE

Address offset: 0x5F8

Controls normal or continuous sample rate

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CC		[80..2047]	Capture and compare value. Sample rate is 16 MHz/CC
B	RW	MODE			Select mode for sample rate control
			Task	0	Rate is controlled from SAMPLE task
			Timers	1	Rate is controlled from local timer (use CC to control the rate)

6.23.9.13 RESULT.PTR

Address offset: 0x62C

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: Only bits 29 and 15..0 are implemented. Writing a value which is not in the 0x2000nnnn range will not read back correctly.

Note: See Memory on page 19 for details about memories available to EasyDMA.

6.23.9.14 RESULT.MAXCNT

Address offset: 0x630

Maximum number of 16-bit samples to be written to output RAM buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT			Maximum number of 16-bit samples to be written to output RAM buffer

6.23.9.15 RESULT.AMOUNT

Address offset: 0x634

Number of 16-bit samples written to output RAM buffer since the previous START task

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT			Number of 16-bit samples written to output RAM buffer since the previous START task. This register can be read after an END or STOPPED event.

6.23.10 Electrical specification

6.23.10.1 SAADC electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
DNL_{10}	Differential non-linearity, 10-bit resolution	-0.95	<1		LSB10b
INL_{10}	Integral non-linearity, 10-bit resolution		1		LSB1(
DNL_{12}	Differential non-linearity, 12-bit resolution	-0.95	1.3		LSB12b
INL_{12}	Integral non-linearity, 12-bit resolution		4.7		LSB1:
V_{OS}	Differential offset error (calibrated), 10-bit resolution [27]		+-2		LSB10b
$E_{VDDHDIV5}$	Error on VDDHDIV5 input		+-1		%
C_{EG}	Gain error temperature coefficient		0.02		%/°C
f_{SAMPLE}	Maximum sampling rate			200	kHz
$t_{ACQ,10k}$	Acquisition time (configurable), source resistance <= 10 kΩ		3		µs
$t_{ACQ,40k}$	Acquisition time (configurable), source resistance <= 40 kΩ		5		µs
$t_{ACQ,100k}$	Acquisition time (configurable), source resistance <= 100 kΩ		10		µs
$t_{ACQ,200k}$	Acquisition time (configurable), source resistance <= 200 kΩ		15		µs
$t_{ACQ,400k}$	Acquisition time (configurable), source resistance <= 400 kΩ		20		µs
$t_{ACQ,800k}$	Acquisition time (configurable), source resistance <= 800 kΩ		40		µs
t_{CONV}	Conversion time		<2		µs
$E_{G1/6}$	Error[28] for gain = 1/6	-3		3	%
$E_{G1/4}$	Error[28] for gain = 1/4	-3		3	%
$E_{G1/2}$	Error[28] for gain = 1/2	-3		4	%
E_{G1}	Error[28] for gain = 1	-3		4	%
C_{SAMPLE}	Sample and hold capacitance at maximum gain[29]		2.5		pF
R_{INPUT}	Input resistance		>1		MΩ
E_{NOB}	Effective number of bits, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		9		Bit
S_{NDR}	Peak signal to noise and distortion ratio, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		56		dB
S_{FDR}	Spurious free dynamic range, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		70		dBc
R_{LADDER}	Ladder resistance		160		kΩ

27 Digital output code at zero volt differential input.
28 Does not include temperature drift
29 Maximum gain corresponds to highest capacitance.

6.24 SPI — Serial peripheral interface master

The SPI master provides a simple CPU interface which includes a TXD register for sending data and an RXD register for receiving data. This section is added for legacy support for now.

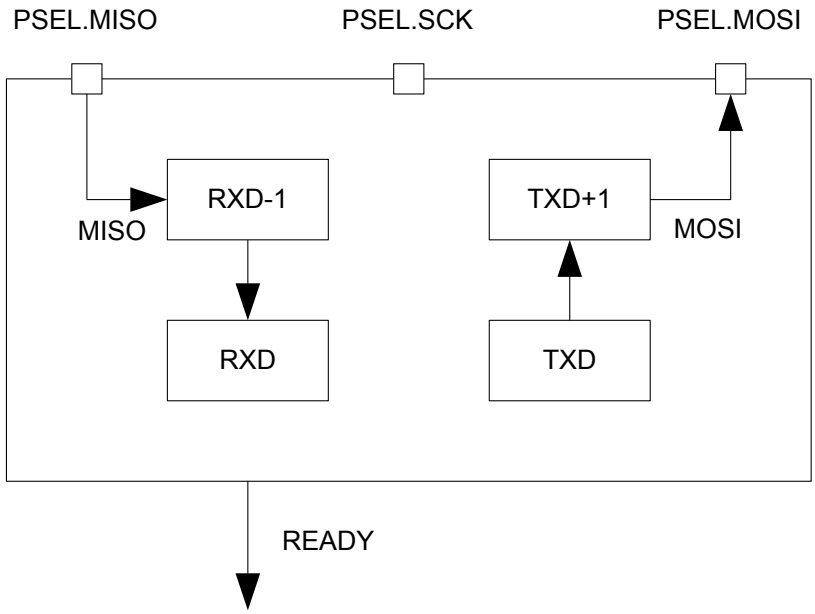

Figure 152: SPI master

RXD-1 and TXD+1 illustrate the double buffered version of RXD and TXD respectively.

6.24.1 Functional description

The TXD and RXD registers are double-buffered to enable some degree of uninterrupted data flow in and out of the SPI master.

The SPI master does not implement support for chip select directly. Therefore, the CPU must use available GPIOs to select the correct slave and control this independently of the SPI master. The SPI master supports SPI modes 0 through 3.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Leading)	0 (Active high)
SPI_MODE1	0 (Leading)	1 (Active low)
SPI_MODE2	1 (Trailing)	0 (Active high)
SPI_MODE3	1 (Trailing)	1 (Active low)

Table 97: SPI modes

6.24.1.1 SPI master mode pin configuration

The different signals SCK, MOSI, and MISO associated with the SPI master are mapped to physical pins.

This mapping is according to the configuration specified in the PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively. If the CONNECT field of a PSEL.xxx register is set to Disconnected, the associated SPI master signal is not connected to any physical pin. The PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI master is enabled, and retained only as long as the device is in ON mode. PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI master is disabled.

To secure correct behavior in the SPI, the pins used by the SPI must be configured in the GPIO peripheral as described in GPIO configuration on page 368 prior to enabling the SPI. The SCK must always be connected to a pin, and that pin's input buffer must always be connected for the SPI to work. This configuration must be retained in the GPIO for the selected IOs as long as the SPI is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time, failing to do so may result in unpredictable behavior.

SPI master signal	SPI master pin	Direction	Output value
SCK	As specified in PSEL.SCK	Output	Same as CONFIG.CPOL
MOSI	As specified in PSEL.MOSI	Output	0
MISO	As specified in PSEL.MISO	Input	Not applicable

Table 98: GPIO configuration

6.24.1.2 Shared resources

The SPI shares registers and other resources with other peripherals that have the same ID as the SPI. Therefore, the user must disable all peripherals that have the same ID as the SPI before the SPI can be configured and used.

Disabling a peripheral that has the same ID as the SPI will not reset any of the registers that are shared with the SPI. It is therefore important to configure all relevant SPI registers explicitly to secure that it operates correctly.

See the Instantiation table in Instantiation on page 22 for details on peripherals and their IDs.

6.24.1.3 SPI master transaction sequence

An SPI master transaction is started by writing the first byte, which is to be transmitted by the SPI master, to the TXD register.

Since the transmitter is double buffered, the second byte can be written to the TXD register immediately after the first one. The SPI master will then send these bytes in the order they are written to the TXD register.

The SPI master is a synchronous interface, and for every byte that is sent, a different byte will be received at the same time; this is illustrated in SPI master transaction on page 369. Bytes that are received will be moved to the RXD register where the CPU can extract them by reading the register. The RXD register is double buffered in the same way as the TXD register, and a second byte can therefore be received at the same time as the first byte is being extracted from RXD by the CPU. The SPI master will generate a READY event every time a new byte is moved to the RXD register. The double buffered byte will be moved from RXD-1 to RXD as soon as the first byte is extracted from RXD. The SPI master will stop when there are no more bytes to send in TXD and TXD+1.

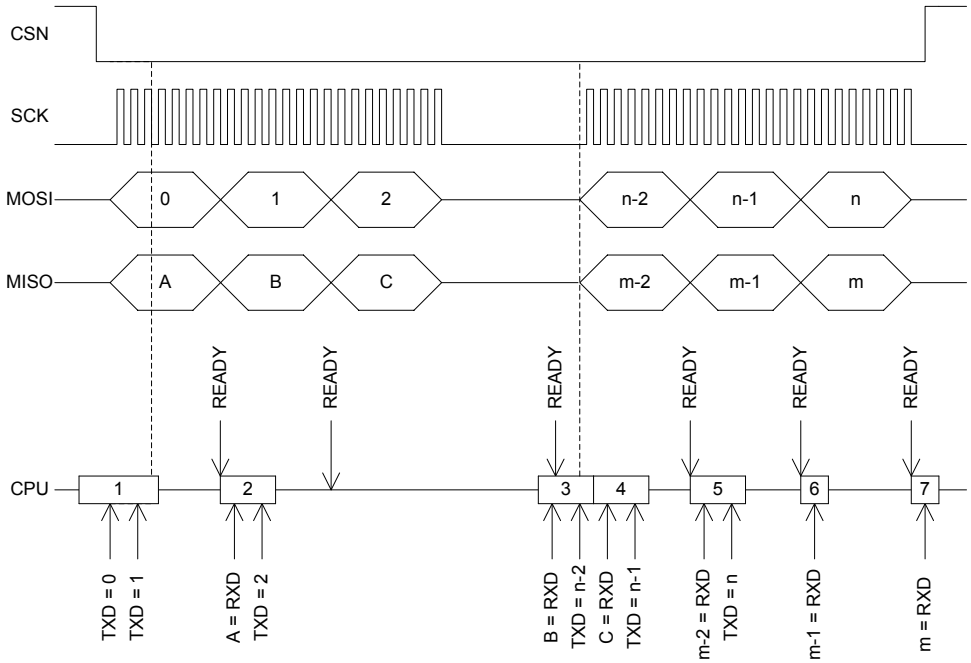

Figure 153: SPI master transaction

The READY event of the third byte transaction is delayed until B is extracted from RXD in occurrence number 3 on the horizontal lifeline. The reason for this is that the third event is generated first when C is moved from RXD-1 to RXD after B is read.

The SPI master will move the incoming byte to the RXD register after a short delay following the SCK clock period of the last bit in the byte. This also means that the READY event will be delayed accordingly, see SPI master transaction on page 369. Therefore, it is important that you always clear the READY event, even if the RXD register and the data that is being received is not used.

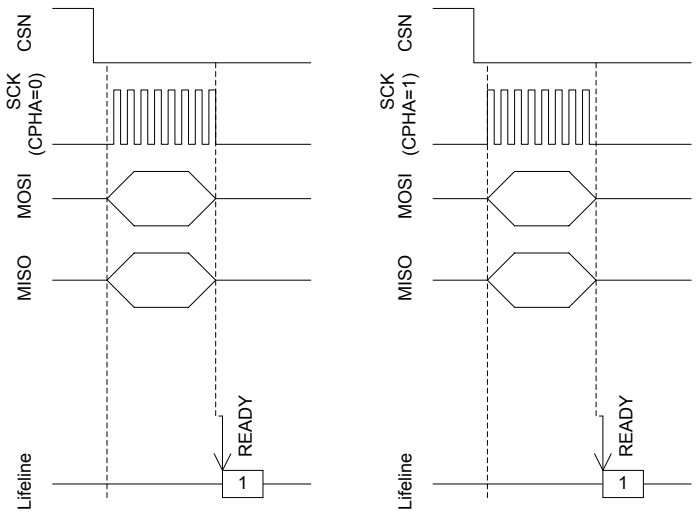

Figure 154: SPI master transaction

6.24.2 Registers

Base address	Peripheral	Instance	Description	Configuration	
0x40003000	SPI	SPI0	SPI master 0		Deprecated
0x40004000	SPI	SPI1	SPI master 1		Deprecated
0x40023000	SPI	SPI2	SPI master 2		Deprecated

Table 99: Instances

Register	Offset	Description
EVENTS_READY	0x108	TXD byte sent and RXD byte received
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable SPI
PSEL.SCK	0x508	Pin select for SCK
PSEL.MOSI	0x50C	Pin select for MOSI signal
PSEL.MISO	0x510	Pin select for MISO signal
RXD	0x518	RXD register
TXD	0x51C	TXD register
FREQUENCY	0x524	SPI frequency. Accuracy depends on the HFCLK source selected.
CONFIG	0x554	Configuration register

Table 100: Register overview

6.24.2.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for READY event See EVENTS_READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.24.2.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for READY event See EVENTS_READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.24.2.3 ENABLE

Address offset: 0x500

Enable SPI

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPI
			Disabled	0	Disable SPI
			Enabled	1	Enable SPI

6.24.2.4 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.24.2.5 PSEL.MOSI

Address offset: 0x50C

Pin select for MOSI signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.24.2.6 PSEL.MISO

Address offset: 0x510

Pin select for MISO signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.24.2.7 RXD

Address offset: 0x518

RXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXD			RX data received. Double buffered

6.24.2.8 TXD

Address offset: 0x51C

TXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TXD			TX data to send. Double buffered

6.24.2.9 FREQUENCY

Address offset: 0x524

SPI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQUENCY			SPI master data rate
			K125	0x02000000	125 kbps
			K250	0x04000000	250 kbps
			K500	0x08000000	500 kbps
			M1	0x10000000	1 Mbps
			M2	0x20000000	2 Mbps
			M4	0x40000000	4 Mbps
			M8	0x80000000	8 Mbps

6.24.2.10 CONFIG

Address offset: 0x554

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ORDER			Bit order
			MsbFirst	0	Most significant bit shifted out first
			LsbFirst	1	Least significant bit shifted out first
B	RW	CPHA			Serial clock (SCK) phase
			Leading	0	Sample on leading edge of clock, shift serial data on trailing edge
			Trailing	1	Sample on trailing edge of clock, shift serial data on leading edge
C	RW	CPOL			Serial clock (SCK) polarity
			ActiveHigh	0	Active high
			ActiveLow	1	Active low

6.24.3 Electrical specification

6.24.3.1 SPI master interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPI}	Bit rates for SPI[30]			8[31]	Mbps
$t_{SPI,START}$	Time from writing TXD register to transmission started		1		µs

6.24.3.2 Serial Peripheral Interface (SPI) Master timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPI,CSCK}$	SCK period	125			ns
$t_{SPI,RSCK,LD}$	SCK rise time, standard drive[a]			$t_{RF,25pF}$	

30 High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

31 The actual maximum data rate depends on the slave's CLK to MISO and MOSI setup and hold timings.

a At 25pF load, including GPIO capacitance, see GPIO spec.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPI,RSCK,HD}$	SCK rise time, high drive[a]			$t_{HRF,25pF}$	
$t_{SPI,FSCK,LD}$	SCK fall time, standard drive[a]			$t_{RF,25pF}$	
$t_{SPI,FSCK,HD}$	SCK fall time, high drive[a]			$t_{HRF,25pF}$	
$t_{SPI,WHSCK}$	SCK high time[a]	$(0.5*t_{CSCK}) - t_{RSCK}$			
$t_{SPI,WLSCK}$	SCK low time[a]	$(0.5*t_{CSCK}) - t_{FSCK}$			
$t_{SPI,SUMI}$	MISO to CLK edge setup time	19			ns
$t_{SPI,HMI}$	CLK edge to MISO hold time	18			ns
$t_{SPI,VMO}$	CLK edge to MOSI valid			59	ns
$t_{SPI,HMO}$	MOSI hold time after CLK edge	20			ns

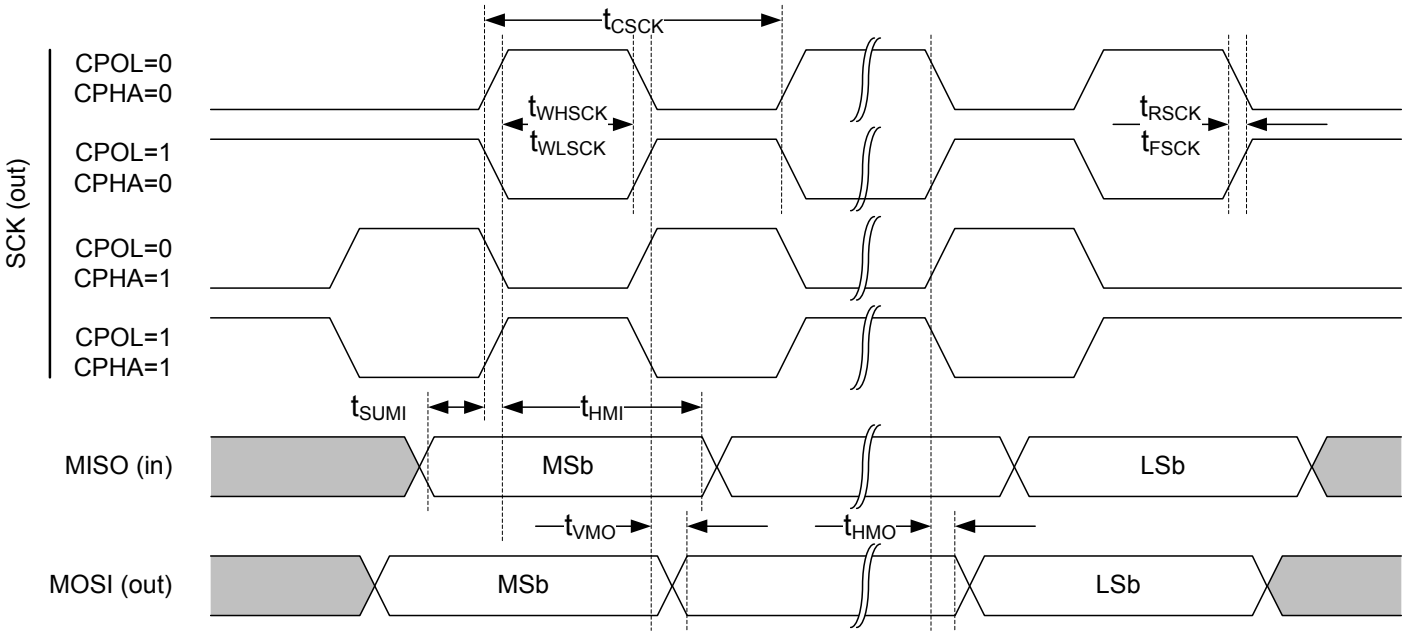

Figure 155: SPI master timing diagram

6.25 SPIM — Serial peripheral interface master with EasyDMA

The SPI master can communicate with multiple SPI slaves using individual chip select signals for each slave.

Listed here are the main features for the SPIM

- EasyDMA direct transfer to/from RAM
- SPI mode 0-3
- Individual selection of I/O pins
- Optional D/CX output line for distinguishing between command and data bytes

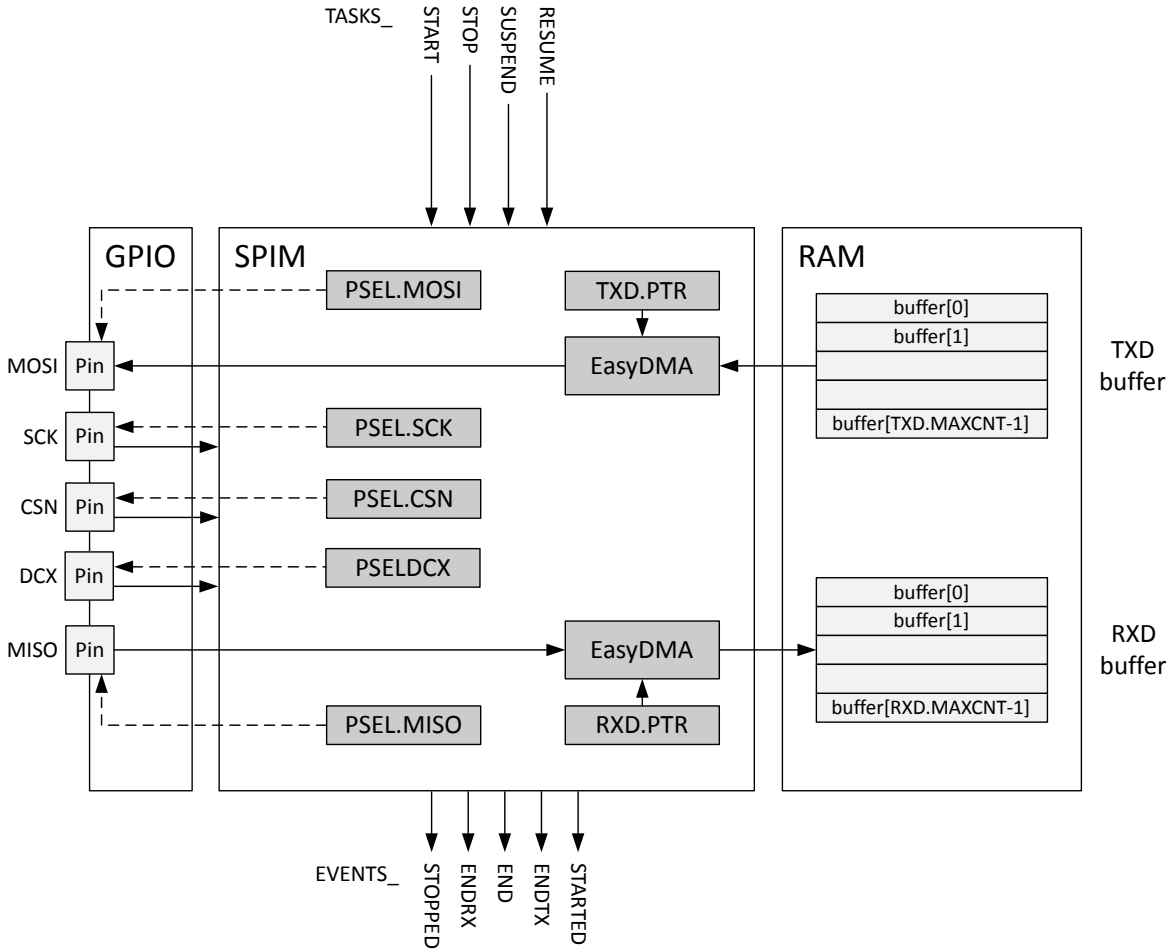

Figure 156: SPIM — SPI master with EasyDMA

6.25.1 SPI master transaction sequence

An SPI master transaction is started by triggering the START task. When started, a number of bytes will be transmitted/received on MOSI/MISO.

The following figure illustrates an SPI master transaction:

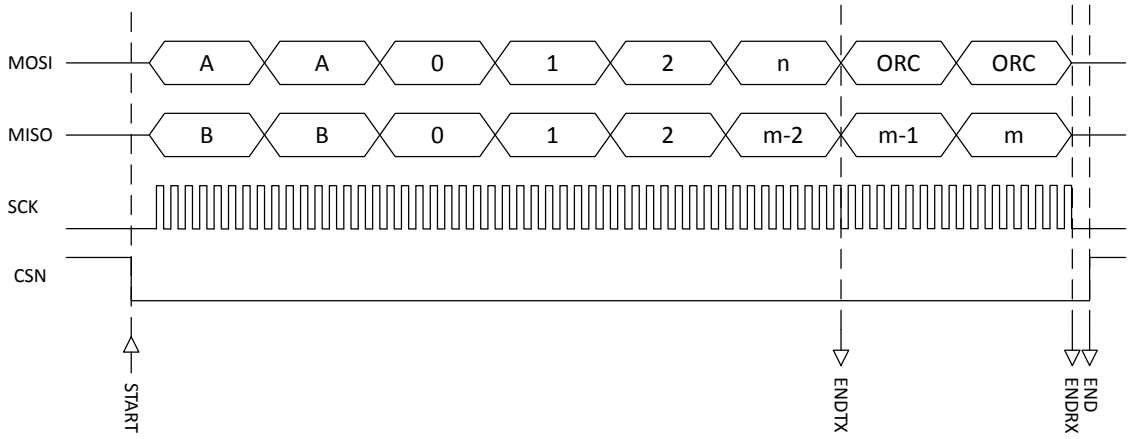

Figure 157: SPI master transaction

The ENDTX is generated when all bytes in buffer TXD.PTR on page 385 are transmitted. The number of bytes in the transmit buffer is specified in register TXD.MAXCNT on page 385. The ENDRX event will be generated when buffer RXD.PTR on page 384 is full, that is when the number of bytes specified in register RXD.MAXCNT on page 384 have been received. The transaction stops automatically after all bytes have been transmitted/received. When the maximum number of bytes in receive buffer is larger than the number of bytes in the transmit buffer, the contents of register ORC on page 388 will be transmitted after the last byte in the transmit buffer has been transmitted.

The END event will be generated after both the ENDRX and ENDTX events have been generated.

The SPI master can be stopped in the middle of a transaction by triggering the STOP task. When triggering the STOP task the SPIM will complete the transmission/reception of the current byte before stopping. A STOPPED event is generated when the SPI master has stopped.

If the ENDTX event has not already been generated when the SPI master has come to a stop, the ENDTX event will be generated even if all bytes in the buffer TXD.PTR on page 385 have not been transmitted.

If the ENDRX event has not already been generated when the SPI master has come to a stop, the ENDRX event will be generated even if the buffer RXD.PTR on page 384 is not full.

A transaction can be suspended and resumed using the SUSPEND and RESUME tasks, receptively. When the SUSPEND task is triggered the SPI master will complete transmitting and receiving the current ongoing byte before it is suspended.

6.25.2 D/CX functionality

Some SPI slaves, for example display drivers, require an additional signal from the SPI master to distinguish between command and data bytes. For display drivers this line is often called D/CX.

The SPIM provides support for such a D/CX output line. The D/CX line is set low during transmission of command bytes and high during transmission of data bytes.

The D/CX pin number is selected using PSELDCX on page 387 and the number of command bytes preceding the data bytes is configured using DCXCNT on page 387.

It is not allowed to write to the DCXCNT on page 387 during an ongoing transmission.

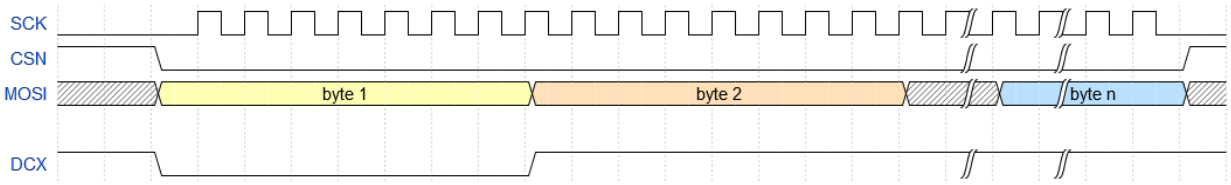

Figure 158: D/CX example. SPIM.DCXCNT = 1.

6.25.3 Pin configuration

The SCK, CSN, DCX, MOSI, and MISO signals associated with the SPIM are mapped to physical pins according to the configuration specified in the PSEL.n registers.

The contents of registers PSEL.SCK on page 382, PSEL.CSN on page 383, PSELDCX on page 387, PSEL.MOSI on page 382 and PSEL.MISO on page 383 are only used when the SPIM is enabled and retained only as long as the device is in System ON mode. The PSEL.n registers can only be configured when the SPIM is disabled. Enabling/disabling is done using register ENABLE on page 382.

To ensure correct behavior, the pins used by the SPIM must be configured in the GPIO peripheral as described in GPIO configuration on page 377 before the SPIM is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI master signal	SPI master pin	Direction	Output value	Comments
SCK	As specified in PSEL.SCK on page 382	Output	Same as CONFIG.CPOL	
CSN	As specified in PSEL.CSN on page 383	Output	Same as CONFIG.CPOL	
DCX	As specified in PSELDCX on page 387	Output	1	
MOSI	As specified in PSEL.MOSI on page 382	Output	0	
MISO	As specified in PSEL.MISO on page 383	Input	Not applicable	

Table 101: GPIO configuration

Some SPIM instances do not support automatic control of CSN, and for those the available GPIO pins need to be used to control CSN directly. See Instances on page 379 for information about what features are supported in the various SPIM instances.

The SPIM supports SPI modes 0 through 3. The clock polarity (CPOL) and the clock phase (CPHA) are configured in register CONFIG on page 386.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Active High)	0 (Leading)
SPI_MODE1	0 (Active High)	1 (Trailing)
SPI_MODE2	1 (Active Low)	0 (Leading)
SPI_MODE3	1 (Active Low)	1 (Trailing)

Table 102: SPI modes

6.25.4 EasyDMA

The SPIM implements EasyDMA for accessing RAM without CPU involvement.

If RXD.PTR and TXD.PTR are not pointing to Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

If several AHB bus masters try to access the same AHB slave at the same time, AHB bus congestion might occur. The SPIM is such an AHB master, and when congestion occurs the SPIM might have to wait for access to RAM to be granted. If the SPIM has to wait for bus access the transaction will be temporary stalled, but resume once the SPIM is granted access to the bus. See the AHB multilayer on page 48 for more information on this subject. Note that some SPIM instances do not support this stalling mechanism. Refer to Instances on page 379 for information about what features are supported in the various instances.

6.25.4.1 EasyDMA array list

The EasyDMA array list can be represented by the data structure ArrayList_type as illustrated in the code example.

```
#define BUFFER_SIZE  4

 typedef struct ArrayList
 {
   uint8_t buffer[BUFFER_SIZE];
 } ArrayList_type;

 ArrayList_type MyArrayList[3];

 //replace 'Channel' below by the specific data channel you want to use,
 //         for instance 'NRF_SPIM->RXD', 'NRF_TWIM->RXD', etc.
 Channel.MAXCNT = BUFFER_SIZE;
 Channel.PTR = &MyArrayList;
```

This data structure includes only a buffer with size equal to Channel.MAXCNT. EasyDMA will use the Channel.MAXCNT register to determine when the buffer is full. Replace 'Channel' by the specific data channel you want to use, for instance 'NRF_SPIM->RXD', 'NRF_SPIM->TXD', 'NRF_TWIM->RXD', etc.

The Channel.MAXCNT register cannot be specified larger than the actual size of the buffer. If Channel.MAXCNT is specified larger than the size of the buffer, the EasyDMA channel may overflow the buffer.

This array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

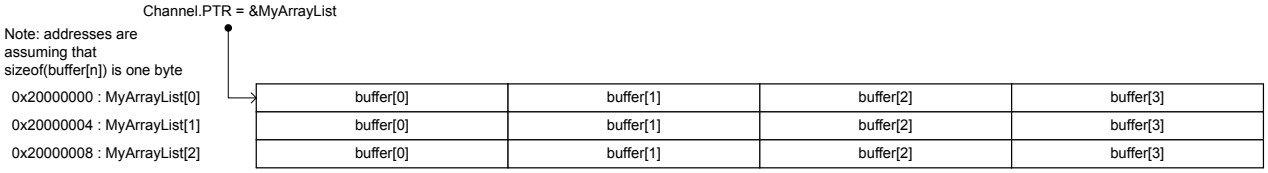

Figure 159: EasyDMA array list

6.25.5 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.25.6 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40003000	SPIM	SPIM0	SPI master 0	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention.
0x40004000	SPIM	SPIM1	SPI master 1	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention.
0x40023000	SPIM	SPIM2	SPI master 2	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention.
0x4002F000	SPIM	SPIM3	SPI master 3	

Table 103: Instances

Register	Offset	Description
TASKS_START	0x010	Start SPI transaction
TASKS_STOP	0x014	Stop SPI transaction
TASKS_SUSPEND	0x01C	Suspend SPI transaction
TASKS_RESUME	0x020	Resume SPI transaction
EVENTS_STOPPED	0x104	SPI transaction has stopped
EVENTS_ENDRX	0x110	End of RXD buffer reached
EVENTS_END	0x118	End of RXD buffer and TXD buffer reached
EVENTS_ENDTX	0x120	End of TXD buffer reached
EVENTS_STARTED	0x14C	Transaction started
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
STALLSTAT	0x400	Stall status for EasyDMA RAM accesses. The fields in this register is set to STALL by hardware whenever a stall occurres and can be cleared (set to NOSTALL) by the CPU.
ENABLE	0x500	Enable SPIM
PSEL.SCK	0x508	Pin select for SCK
PSEL.MOSI	0x50C	Pin select for MOSI signal
PSEL.MISO	0x510	Pin select for MISO signal
PSEL.CSN	0x514	Pin select for CSN
FREQUENCY	0x524	SPI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534	Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last transaction
RXD.LIST	0x540	EasyDMA list type
TXD.PTR	0x544	Data pointer
TXD.MAXCNT	0x548	Number of bytes in transmit buffer

Register	Offset	Description
TXD.AMOUNT	0x54C	Number of bytes transferred in the last transaction
TXD.LIST	0x550	EasyDMA list type
CONFIG	0x554	Configuration register
IFTIMING.RXDELAY	0x560	Sample delay for input serial data on MISO
IFTIMING.CSNDUR	0x564	Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions
CSNPOL	0x568	Polarity of CSN output
PSELDCX	0x56C	Pin select for DCX signal
DCXCNT	0x570	DCX configuration
ORC	0x5C0	Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT

Table 104: Register overview

6.25.6.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END_START			Shortcut between END event and START task See EVENTS_END and TASKS_START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.25.6.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to enable interrupt for ENDRX event See EVENTS_ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
D	RW	ENDTX			Write '1' to enable interrupt for ENDTX event See EVENTS_ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STARTED			Write '1' to enable interrupt for STARTED event See EVENTS_STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.25.6.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to disable interrupt for ENDRX event See EVENTS_ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to disable interrupt for END event See EVENTS_END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDTX			Write '1' to disable interrupt for ENDTX event See EVENTS_ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STARTED			Write '1' to disable interrupt for STARTED event See EVENTS_STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.25.6.4 STALLSTAT

Address offset: 0x400

Stall status for EasyDMA RAM accesses. The fields in this register is set to STALL by hardware whenever a stall occurres and can be cleared (set to NOSTALL) by the CPU.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TX		[1..0]	Stall status for EasyDMA RAM reads
			NOSTALL	0	No stall
			STALL	1	A stall has occurred
B	RW	RX		[1..0]	Stall status for EasyDMA RAM writes
			NOSTALL	0	No stall
			STALL	1	A stall has occurred

6.25.6.5 ENABLE

Address offset: 0x500

Enable SPIM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPIM
			Disabled	0	Disable SPIM
			Enabled	7	Enable SPIM

6.25.6.6 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.25.6.7 PSEL.MOSI

Address offset: 0x50C

Pin select for MOSI signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.25.6.8 PSEL.MISO

Address offset: 0x510

Pin select for MISO signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.25.6.9 PSEL.CSN

Address offset: 0x514

Pin select for CSN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.25.6.10 FREQUENCY

Address offset: 0x524

SPI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQUENCY			SPI master data rate
			K125	0x02000000	125 kbps
			K250	0x04000000	250 kbps
			K500	0x08000000	500 kbps
			M1	0x10000000	1 Mbps
			M2	0x20000000	2 Mbps
			M4	0x40000000	4 Mbps
			M8	0x80000000	8 Mbps
			M16	0x0A000000	16 Mbps
			M32	0x14000000	32 Mbps

6.25.6.11 RXD.PTR

Address offset: 0x534

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.25.6.12 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xffff]	Maximum number of bytes in receive buffer

6.25.6.13 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xffff]	Number of bytes transferred in the last transaction

6.25.6.14 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.25.6.15 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.25.6.16 TXD.MAXCNT

Address offset: 0x548

Number of bytes in transmit buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xffff]	Maximum number of bytes in transmit buffer

6.25.6.17 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xffff]	Number of bytes transferred in the last transaction

6.25.6.18 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.25.6.19 CONFIG

Address offset: 0x554

Configuration register

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ORDER			Bit order
			MsbFirst	0	Most significant bit shifted out first
			LsbFirst	1	Least significant bit shifted out first
B	RW	CPHA			Serial clock (SCK) phase
			Leading	0	Sample on leading edge of clock, shift serial data on trailing edge
			Trailing	1	Sample on trailing edge of clock, shift serial data on leading edge
C	RW	CPOL			Serial clock (SCK) polarity
			ActiveHigh	0	Active high
			ActiveLow	1	Active low

6.25.6.20 IFTIMING.RXDELAY

Address offset: 0x560

Sample delay for input serial data on MISO

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A
Reset 0x00000002					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RXDELAY		[7..0]	Sample delay for input serial data on MISO. The value specifies the number of 64 MHz clock cycles (15.625 ns) delay from the the sampling edge of SCK (leading edge for CONFIG.CPHA = 0, trailing edge for CONFIG.CPHA = 1) until the input serial data is sampled. As en example, if RXDELAY = 0 and CONFIG.CPHA = 0, the input serial data is sampled on the rising edge of SCK.

6.25.6.21 IFTIMING.CSNDUR

Address offset: 0x564

Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CSNDUR		[0xFF..0]	Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions. The value is specified in number of 64 MHz clock cycles (15.625 ns).

6.25.6.22 CSNPOL

Address offset: 0x568

Polarity of CSN output

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CSNPOL			Polarity of CSN output
			LOW	0	Active low (idle state high)
			HIGH	1	Active high (idle state low)

6.25.6.23 PSELDCX

Address offset: 0x56C

Pin select for DCX signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.25.6.24 DCXCNT

Address offset: 0x570

DCX configuration

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	DCXCNT		0x0..0xF	This register specifies the number of command bytes preceding the data bytes. The PSEL.DCX line will be low during transmission of command bytes and high during transmission of data bytes. Value 0xF indicates that all bytes are command bytes.

6.25.6.25 ORC

Address offset: 0x5C0

Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ORC			Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT.

6.25.7 Electrical specification

6.25.7.1 Timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIM}	Bit rates for SPIM[32]			32	Mbps
$t_{SPIM,START}$	Time from START task to transmission started		1		µs
$t_{SPIM,CSCK}$	SCK period	125			ns
$t_{SPIM,RSCK,LD}$	SCK rise time, standard drive[33]			$t_{RF,25pF}$	
$t_{SPIM,RSCK,HD}$	SCK rise time, high drive[33]			$t_{HRF,25pF}$	
$t_{SPIM,FSCK,LD}$	SCK fall time, standard drive[33]			$t_{RF,25pF}$	
$t_{SPIM,FSCK,HD}$	SCK fall time, high drive[33]			$t_{HRF,25pF}$	
$t_{SPIM,WHSCK}$	SCK high time[33]	$(0.5*t_{CSCK} - t_{RSCK}$			
$t_{SPIM,WLSCK}$	SCK low time[33]	$(0.5*t_{CSCK}) - t_{FSCK}$			
$t_{SPIM,SUMI}$	MISO to CLK edge setup time	19			ns
$t_{SPIM,HMI}$	CLK edge to MISO hold time	18			ns
$t_{SPIM,VMO}$	CLK edge to MOSI valid, SCK frequency <= 8 MHz			59	ns
$t_{SPIM,VMO,HS}$	CLK edge to MOSI valid, SCK frequency > 8 MHz			8	ns
$t_{SPIM,HMO}$	MOSI hold time after CLK edge	20			ns

[32] High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.
[33] At 25pF load, including GPIO pin capacitance, see GPIO spec.

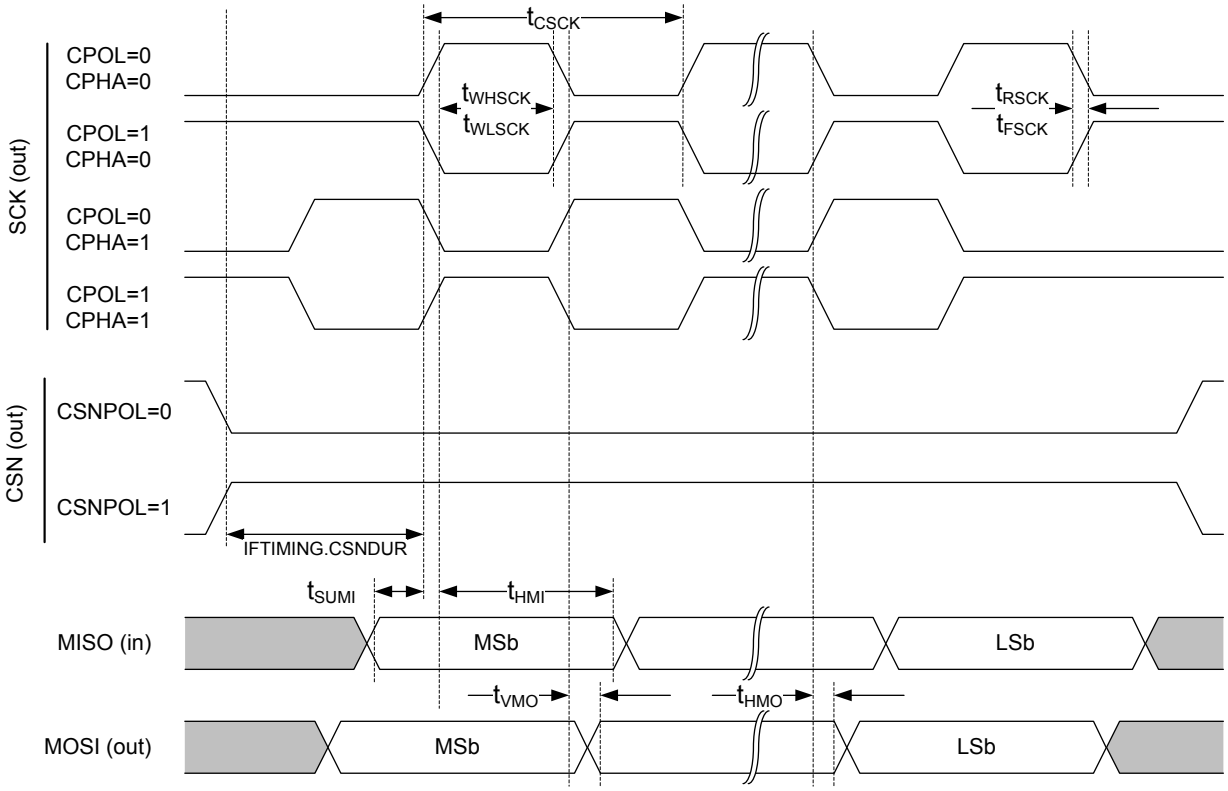

Figure 160: SPIM timing diagram

6.26 SPIS — Serial peripheral interface slave with EasyDMA

SPI slave (SPIS) is implemented with EasyDMA support for ultra low power serial communication from an external SPI master. EasyDMA in conjunction with hardware-based semaphore mechanisms removes all real-time requirements associated with controlling the SPI slave from a low priority CPU execution context.

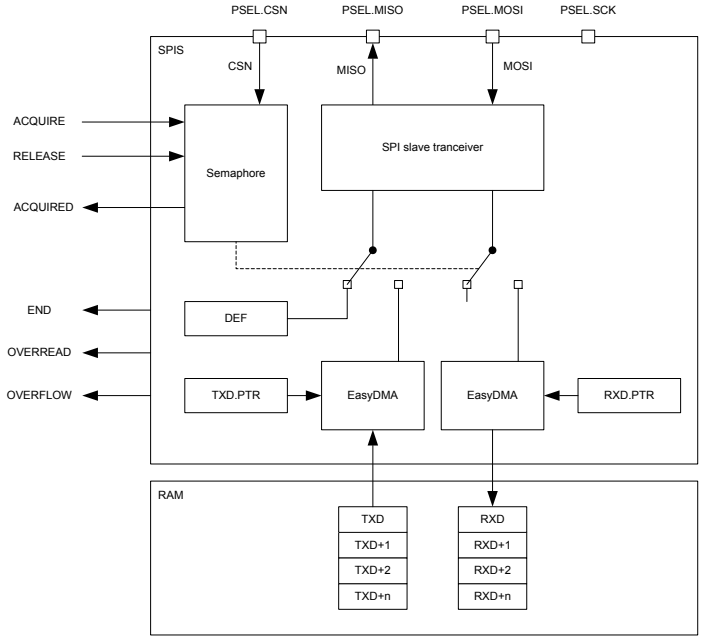

Figure 161: SPI slave

The SPIS supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Leading)	0 (Active High)
SPI_MODE1	0 (Leading)	1 (Active Low)
SPI_MODE2	1 (Trailing)	0 (Active High)
SPI_MODE3	1 (Trailing)	1 (Active Low)

Table 105: SPI modes

6.26.1 Shared resources

The SPI slave shares registers and other resources with other peripherals that have the same ID as the SPI slave. Therefore, you must disable all peripherals that have the same ID as the SPI slave before the SPI slave can be configured and used.

Disabling a peripheral that has the same ID as the SPI slave will not reset any of the registers that are shared with the SPI slave. It is important to configure all relevant SPI slave registers explicitly to secure that it operates correctly.

The Instantiation table in Instantiation on page 22 shows which peripherals have the same ID as the SPI slave.

6.26.2 EasyDMA

The SPI slave implements EasyDMA for reading and writing to and from the RAM. The END event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

6.26.3 SPI slave operation

SPI slave uses two memory pointers, RXD.PTR and TXD.PTR, that point to the RXD buffer (receive buffer) and TXD buffer (transmit buffer) respectively. Since these buffers are located in RAM, which can be accessed by both the SPI slave and the CPU, a hardware based semaphore mechanism is implemented to enable safe sharing.

See SPI transaction when shortcut between END and ACQUIRE is enabled on page 392.

Before the CPU can safely update the RXD.PTR and TXD.PTR pointers it must first acquire the SPI semaphore. The CPU can acquire the semaphore by triggering the ACQUIRE task and then receiving the ACQUIRED event. When the CPU has updated the RXD.PTR and TXD.PTR pointers the CPU must release the semaphore before the SPI slave will be able to acquire it. The CPU releases the semaphore by triggering the RELEASE task. This is illustrated in SPI transaction when shortcut between END and ACQUIRE is enabled on page 392. Triggering the RELEASE task when the semaphore is not granted to the CPU will have no effect.

The semaphore mechanism does not, at any time, prevent the CPU from performing read or write access to the RXD.PTR register, the TXD.PTR registers, or the RAM that these pointers are pointing to. The semaphore is only telling when these can be updated by the CPU so that safe sharing is achieved.

The semaphore is by default assigned to the CPU after the SPI slave is enabled. No ACQUIRED event will be generated for this initial semaphore handover. An ACQUIRED event will be generated immediately if the ACQUIRE task is triggered while the semaphore is assigned to the CPU.

The SPI slave will try to acquire the semaphore when CSN goes low. If the SPI slave does not manage to acquire the semaphore at this point, the transaction will be ignored. This means that all incoming data on MOSI will be discarded, and the DEF (default) character will be clocked out on the MISO line throughout the whole transaction. This will also be the case even if the semaphore is released by the CPU during the transaction. In case of a race condition where the CPU and the SPI slave try to acquire the semaphore at the same time, as illustrated in lifeline item 2 in SPI transaction when shortcut between END and ACQUIRE is enabled on page 392, the semaphore will be granted to the CPU.

If the SPI slave acquires the semaphore, the transaction will be granted. The incoming data on MOSI will be stored in the RXD buffer and the data in the TXD buffer will be clocked out on MISO.

When a granted transaction is completed and CSN goes high, the SPI slave will automatically release the semaphore and generate the END event.

As long as the semaphore is available the SPI slave can be granted multiple transactions one after the other. If the CPU is not able to reconfigure the TXD.PTR and RXD.PTR between granted transactions, the same TX data will be clocked out and the RX buffers will be overwritten. To prevent this from happening, the END_ACQUIRE shortcut can be used. With this shortcut enabled the semaphore will be handed over to the CPU automatically after the granted transaction has completed, giving the CPU the ability to update the TXPTR and RXPTR between every granted transaction.

If the CPU tries to acquire the semaphore while it is assigned to the SPI slave, an immediate handover will not be granted. However, the semaphore will be handed over to the CPU as soon as the SPI slave has released the semaphore after the granted transaction is completed. If the END_ACQUIRE shortcut is enabled and the CPU has triggered the ACQUIRE task during a granted transaction, only one ACQUIRE request will be served following the END event.

The MAXRX register specifies the maximum number of bytes the SPI slave can receive in one granted transaction. If the SPI slave receives more than MAXRX number of bytes, an OVERFLOW will be indicated in the STATUS register and the incoming bytes will be discarded.

The MAXTX parameter specifies the maximum number of bytes the SPI slave can transmit in one granted transaction. If the SPI slave is forced to transmit more than MAXTX number of bytes, an OVERREAD will be indicated in the STATUS register and the ORC character will be clocked out.

The RXD.AMOUNT and TXD.AMOUNT registers are updated when a granted transaction is completed. The TXD.AMOUNT register indicates how many bytes were read from the TX buffer in the last transaction, that is, ORC (over-read) characters are not included in this number. Similarly, the RXD.AMOUNT register indicates how many bytes were written into the RX buffer in the last transaction.

The ENDRX event is generated when the RX buffer has been filled.

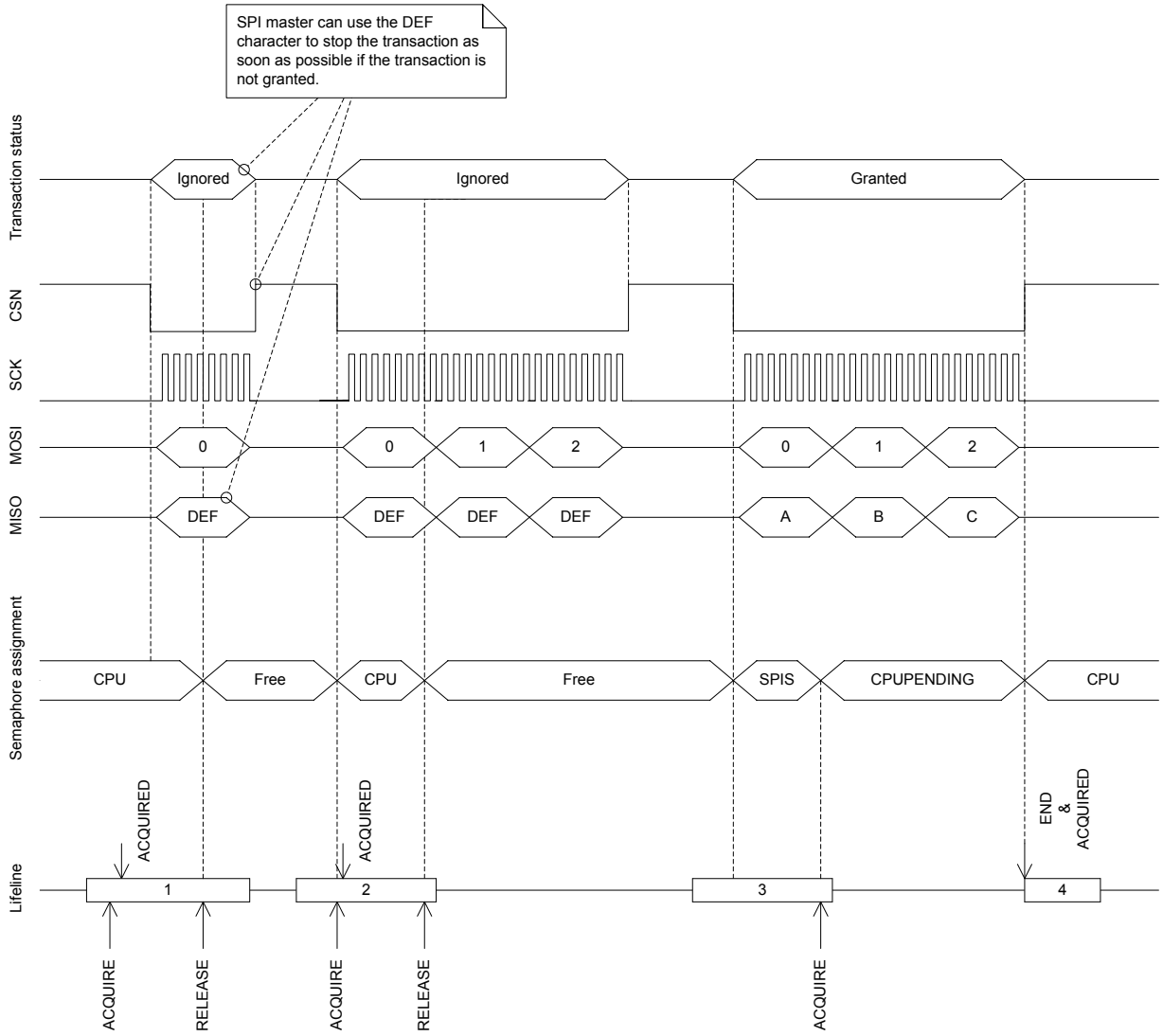

Figure 162: SPI transaction when shortcut between END and ACQUIRE is enabled

6.26.4 Pin configuration

The CSN, SCK, MOSI, and MISO signals associated with the SPI slave are mapped to physical pins according to the configuration specified in the PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively. If the CONNECT field of any of these registers is set to Disconnected, the associated SPI slave signal will not be connected to any physical pins.

The PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI slave is enabled, and retained only as long as the device is in System ON mode, see POWER — Power supply on page 60 chapter for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI slave is disabled.

To secure correct behavior in the SPI slave, the pins used by the SPI slave must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 393 before enabling the SPI slave. This is to secure that the pins used by the SPI slave are driven correctly if the SPI slave itself is temporarily disabled, or if the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the SPI slave is to be recognized by an external SPI master.

The MISO line is set in high impedance as long as the SPI slave is not selected with CSN.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI signal	SPI pin	Direction	Output value	Comment
CSN	As specified in PSEL.CSN	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
MOSI	As specified in PSEL.MOSI	Input	Not applicable	
MISO	As specified in PSEL.MISO	Input	Not applicable	Emulates that the SPI slave is not selected.

Table 106: GPIO configuration before enabling peripheral

6.26.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40003000	SPIS	SPIS0	SPI slave 0	
0x40004000	SPIS	SPIS1	SPI slave 1	
0x40023000	SPIS	SPIS2	SPI slave 2	

Table 107: Instances

Register	Offset	Description	
TASKS_ACQUIRE	0x024	Acquire SPI semaphore	
TASKS_RELEASE	0x028	Release SPI semaphore, enabling the SPI slave to acquire it	
EVENTS_END	0x104	Granted transaction completed	
EVENTS_ENDRX	0x110	End of RXD buffer reached	
EVENTS_ACQUIRED	0x128	Semaphore acquired	
SHORTS	0x200	Shortcut register	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
SEMSTAT	0x400	Semaphore status register	
STATUS	0x440	Status from last transaction	
ENABLE	0x500	Enable SPI slave	
PSEL.SCK	0x508	Pin select for SCK	
PSEL.MISO	0x50C	Pin select for MISO signal	
PSEL.MOSI	0x510	Pin select for MOSI signal	
PSEL.CSN	0x514	Pin select for CSN signal	
PSELSCK	0x508	Pin select for SCK	Deprecated
PSELMISO	0x50C	Pin select for MISO	Deprecated
PSELMOSI	0x510	Pin select for MOSI	Deprecated
PSELCSN	0x514	Pin select for CSN	Deprecated
RXD.PTR	0x534	RXD data pointer	
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer	
RXD.AMOUNT	0x53C	Number of bytes received in last granted transaction	
RXDPTR	0x534	RXD data pointer	Deprecated
MAXRX	0x538	Maximum number of bytes in receive buffer	Deprecated
AMOUNTRX	0x53C	Number of bytes received in last granted transaction	Deprecated
TXD.PTR	0x544	TXD data pointer	
TXD.MAXCNT	0x548	Maximum number of bytes in transmit buffer	
TXD.AMOUNT	0x54C	Number of bytes transmitted in last granted transaction	
TXDPTR	0x544	TXD data pointer	Deprecated
MAXTX	0x548	Maximum number of bytes in transmit buffer	Deprecated
AMOUNTTX	0x54C	Number of bytes transmitted in last granted transaction	Deprecated
CONFIG	0x554	Configuration register	

Register	Offset	Description
DEF	0x55C	Default character. Character clocked out in case of an ignored transaction.
ORC	0x5C0	Over-read character

Table 108: Register overview

6.26.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END_ACQUIRE			Shortcut between END event and ACQUIRE task See EVENTS_END and TASKS_ACQUIRE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.26.5.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	END			Write '1' to enable interrupt for END event See EVENTS_END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to enable interrupt for ENDRX event See EVENTS_ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACQUIRED			Write '1' to enable interrupt for ACQUIRED event See EVENTS_ACQUIRED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.26.5.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	END			Write '1' to disable interrupt for END event See EVENTS_END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to disable interrupt for ENDRX event See EVENTS_ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACQUIRED			Write '1' to disable interrupt for ACQUIRED event See EVENTS_ACQUIRED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.26.5.4 SEMSTAT

Address offset: 0x400

Semaphore status register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	RW	Field	Value ID	Value	Description
A	R	SEMSTAT			Semaphore status
			Free	0	Semaphore is free
			CPU	1	Semaphore is assigned to CPU
			SPIS	2	Semaphore is assigned to SPI slave
			CPUPending	3	Semaphore is assigned to SPI but a handover to the CPU is pending

6.26.5.5 STATUS

Address offset: 0x440

Status from last transaction

Individual bits are cleared by writing a '1' to the bits that shall be cleared

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERREAD			TX buffer over-read detected, and prevented
			NotPresent	0	Read: error not present
			Present	1	Read: error present
			Clear	1	Write: clear error on writing '1'
B	RW	OVERFLOW			RX buffer overflow detected, and prevented
			NotPresent	0	Read: error not present
			Present	1	Read: error present
			Clear	1	Write: clear error on writing '1'

6.26.5.6 ENABLE

Address offset: 0x500

Enable SPI slave

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPI slave
			Disabled	0	Disable SPI slave
			Enabled	2	Enable SPI slave

6.26.5.7 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.26.5.8 PSEL.MISO

Address offset: 0x50C

Pin select for MISO signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.26.5.9 PSEL.MOSI

Address offset: 0x510

Pin select for MOSI signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.26.5.10 PSEL.CSN

Address offset: 0x514

Pin select for CSN signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.26.5.11 PSELSCK (Deprecated)

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELSCK		[0..31]	Pin number configuration for SPI SCK signal
			Disconnected	0xFFFFFFFF	Disconnect

6.26.5.12 PSELMISO (Deprecated)

Address offset: 0x50C

Pin select for MISO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELMISO		[0..31]	Pin number configuration for SPI MISO signal
			Disconnected	0xFFFFFFFF	Disconnect

6.26.5.13 PSELMOSI (Deprecated)

Address offset: 0x510

Pin select for MOSI

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELMOSI		[0..31]	Pin number configuration for SPI MOSI signal
			Disconnected	0xFFFFFFFF	Disconnect

6.26.5.14 PSELCSN (Deprecated)

Address offset: 0x514

Pin select for CSN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PSELCSN		[0..31]	Pin number configuration for SPI CSN signal
			Disconnected	0xFFFFFFFF	Disconnect

6.26.5.15 RXD.PTR

Address offset: 0x534

RXD data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			RXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.26.5.16 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xffff]	Maximum number of bytes in receive buffer

6.26.5.17 RXD.AMOUNT

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xffff]	Number of bytes received in the last granted transaction

6.26.5.18 RXDPTR (Deprecated)

Address offset: 0x534

RXD data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	RXDPTR			RXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.26.5.19 MAXRX (Deprecated)

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXRX		[1..0xffff]	Maximum number of bytes in receive buffer

6.26.5.20 AMOUNTRX (Deprecated)

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNTRX		[1..0xffff]	Number of bytes received in the last granted transaction

6.26.5.21 TXD.PTR

Address offset: 0x544

TXD data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			TXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.26.5.22 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xffff]	Maximum number of bytes in transmit buffer

6.26.5.23 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xffff]	Number of bytes transmitted in last granted transaction

6.26.5.24 TXDPTR (Deprecated)

Address offset: 0x544

TXD data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	TXDPTR			TXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.26.5.25 MAXTX (Deprecated)

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXTX		[1..0xffff]	Maximum number of bytes in transmit buffer

6.26.5.26 AMOUNTTX (Deprecated)

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNTTX		[1..0xffff]	Number of bytes transmitted in last granted transaction

6.26.5.27 CONFIG

Address offset: 0x554

Configuration register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ORDER			Bit order
			MsbFirst	0	Most significant bit shifted out first
			LsbFirst	1	Least significant bit shifted out first
B	RW	CPHA			Serial clock (SCK) phase
			Leading	0	Sample on leading edge of clock, shift serial data on trailing edge
			Trailing	1	Sample on trailing edge of clock, shift serial data on leading edge
C	RW	CPOL			Serial clock (SCK) polarity
			ActiveHigh	0	Active high
			ActiveLow	1	Active low

6.26.5.28 DEF

Address offset: 0x55C

Default character. Character clocked out in case of an ignored transaction.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DEF			Default character. Character clocked out in case of an ignored transaction.

6.26.5.29 ORC

Address offset: 0x5C0

Over-read character

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character clocked out after an over-read of the transmit buffer.

6.26.6 Electrical specification

6.26.6.1 SPIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIS}	Bit rates for SPIS[34]			8[35]	Mbps
$t_{SPIS,START}$	Time from RELEASE task to receive/transmit (CSN active)		0.125		µs

6.26.6.2 Serial Peripheral Interface Slave (SPIS) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,CSCKIN}$	SCK input period	125			ns
$t_{SPIS,RFSCKIN}$	SCK input rise/fall time			30	ns
$t_{SPIS,WHSCKIN}$	SCK input high time	30			ns
$t_{SPIS,WLSCKIN}$	SCK input low time	30			ns
$t_{SPIS,SUCSN}$	CSN to CLK setup time	1000			ns
$t_{SPIS,HCSN}$	CLK to CSN hold time	2000			ns
$t_{SPIS,ASA}$	CSN to MISO driven	0			ns
$t_{SPIS,ASO}$	CSN to MISO valid[a]			1000	ns
$t_{SPIS,DISSO}$	CSN to MISO disabled[a]			68	ns
$t_{SPIS,CWH}$	CSN inactive time	300			ns

34 High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

35 The actual maximum data rate depends on the master's CLK to MISO and MOSI setup and hold timings.

a At 25pF load, including GPIO capacitance, see GPIO spec.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,VSO}$	CLK edge to MISO valid			19	ns
$t_{SPIS,HSO}$	MISO hold time after CLK edge	18[36]			ns
$t_{SPIS,SUSI}$	MOSI to CLK edge setup time	59			ns
$t_{SPIS,HSI}$	CLK edge to MOSI hold time	20			ns

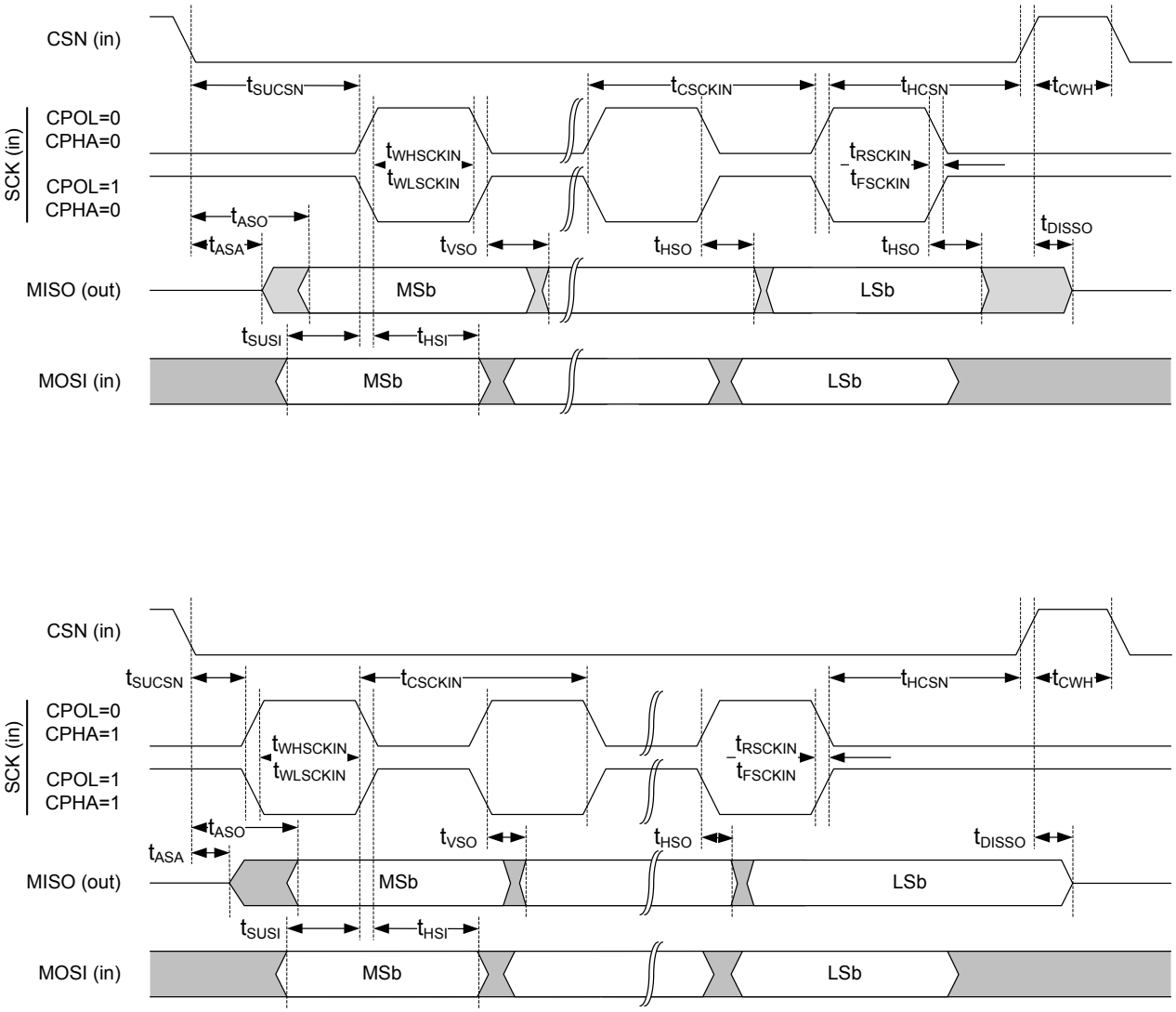

Figure 163: SPIS timing diagram

[36] This is to ensure compatibility to SPI masters sampling MISO on the same edge as MOSI is output

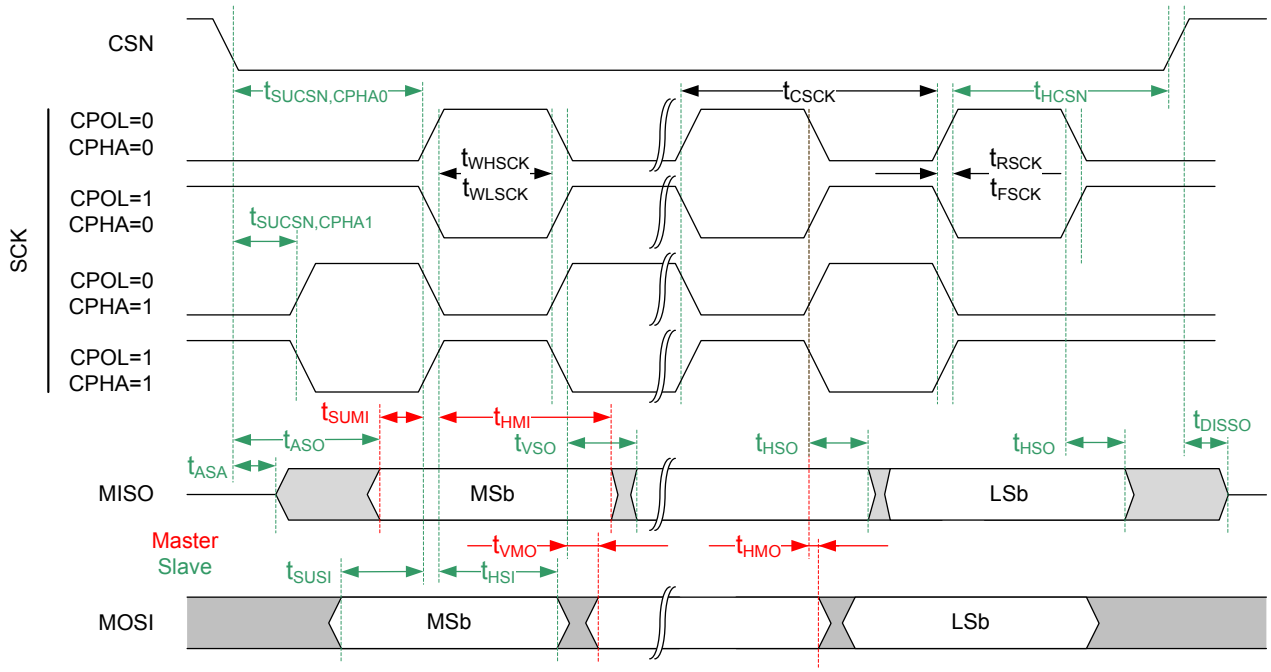

Figure 164: Common SPIM and SPIS timing diagram

6.27 SWI — Software interrupts

A set of interrupts have been reserved for use as software interrupts.

6.27.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40014000	SWI	SWI0	Software interrupt 0	
0x40015000	SWI	SWI1	Software interrupt 1	
0x40016000	SWI	SWI2	Software interrupt 2	
0x40017000	SWI	SWI3	Software interrupt 3	
0x40018000	SWI	SWI4	Software interrupt 4	
0x40019000	SWI	SWI5	Software interrupt 5	

Table 109: Instances

6.28 TEMP — Temperature sensor

The temperature sensor measures die temperature over the temperature range of the device. Linearity compensation can be implemented if required by the application.

Listed here are the main features for TEMP:

- Temperature range is greater than or equal to operating temperature of the device
- Resolution is 0.25 degrees

TEMP is started by triggering the START task.

When the temperature measurement is completed, a DATARDY event will be generated and the result of the measurement can be read from the TEMP register.

To achieve the measurement accuracy stated in the electrical specification, the crystal oscillator must be selected as the HFCLK source, see CLOCK — Clock control on page 80 for more information.

When the temperature measurement is completed, TEMP analog electronics power down to save power.

TEMP only supports one-shot operation, meaning that every TEMP measurement has to be explicitly started using the START task.

6.28.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000C000	TEMP	TEMP	Temperature sensor	

Table 110: Instances

Register	Offset	Description
TASKS_START	0x000	Start temperature measurement
TASKS_STOP	0x004	Stop temperature measurement
EVENTS_DATARDY	0x100	Temperature measurement complete, data ready
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
TEMP	0x508	Temperature in °C (0.25° steps)
A0	0x520	Slope of 1st piece wise linear function
A1	0x524	Slope of 2nd piece wise linear function
A2	0x528	Slope of 3rd piece wise linear function
A3	0x52C	Slope of 4th piece wise linear function
A4	0x530	Slope of 5th piece wise linear function
A5	0x534	Slope of 6th piece wise linear function
B0	0x540	y-intercept of 1st piece wise linear function
B1	0x544	y-intercept of 2nd piece wise linear function
B2	0x548	y-intercept of 3rd piece wise linear function
B3	0x54C	y-intercept of 4th piece wise linear function
B4	0x550	y-intercept of 5th piece wise linear function
B5	0x554	y-intercept of 6th piece wise linear function
T0	0x560	End point of 1st piece wise linear function
T1	0x564	End point of 2nd piece wise linear function
T2	0x568	End point of 3rd piece wise linear function
T3	0x56C	End point of 4th piece wise linear function
T4	0x570	End point of 5th piece wise linear function

Table 111: Register overview

6.28.1.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DATARDY			Write '1' to enable interrupt for DATARDY event
					See EVENTS_DATARDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.28.1.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	DATARDY			Write '1' to disable interrupt for DATARDY event
					See EVENTS_DATARDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.28.1.3 TEMP

Address offset: 0x508

Temperature in °C (0.25° steps)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	TEMP			Temperature in °C (0.25° steps)
					Result of temperature measurement. Die temperature in °C,
					2's complement format, 0.25 °C steps
					Decision point: DATARDY

6.28.1.4 A0

Address offset: 0x520

Slope of 1st piece wise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A
Reset 0x00000326				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0**
ID	RW	Field	Value ID	Value	Description
A	RW	A0			Slope of 1st piece wise linear function

6.28.1.5 A1

Address offset: 0x524

Slope of 2nd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000348	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	A1			Slope of 2nd piece wise linear function

6.28.1.6 A2

Address offset: 0x528

Slope of 3rd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x000003AA	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	A2			Slope of 3rd piece wise linear function

6.28.1.7 A3

Address offset: 0x52C

Slope of 4th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x0000040E	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	A3			Slope of 4th piece wise linear function

6.28.1.8 A4

Address offset: 0x530

Slope of 5th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x000004BD	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1**

ID	RW	Field	Value ID	Value	Description
A	RW	A4			Slope of 5th piece wise linear function

6.28.1.9 A5

Address offset: 0x534

Slope of 6th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x000005A3	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	A5			Slope of 6th piece wise linear function

6.28.1.10 B0

Address offset: 0x540

y-intercept of 1st piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00003FEF	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	B0			y-intercept of 1st piece wise linear function

6.28.1.11 B1

Address offset: 0x544

y-intercept of 2nd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00003FBE	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	B1			y-intercept of 2nd piece wise linear function

6.28.1.12 B2

Address offset: 0x548

y-intercept of 3rd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00003FBE	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	B2			y-intercept of 3rd piece wise linear function

6.28.1.13 B3

Address offset: 0x54C

y-intercept of 4th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000012	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0**

ID	RW	Field	Value ID	Value	Description
A	RW	B3			y-intercept of 4th piece wise linear function

6.28.1.14 B4

Address offset: 0x550

y-intercept of 5th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000124	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	B4			y-intercept of 5th piece wise linear function

6.28.1.15 B5

Address offset: 0x554

y-intercept of 6th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																			A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x0000027C	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	B5			y-intercept of 6th piece wise linear function

6.28.1.16 T0

Address offset: 0x560

End point of 1st piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x000000E2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	T0			End point of 1st piece wise linear function

6.28.1.17 T1

Address offset: 0x564

End point of 2nd piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	T1			End point of 2nd piece wise linear function

6.28.1.18 T2

Address offset: 0x568

End point of 3rd piece wise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000019				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	T2			End point of 3rd piece wise linear function

6.28.1.19 T3

Address offset: 0x56C

End point of 4th piece wise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x0000003C				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	T3			End point of 4th piece wise linear function

6.28.1.20 T4

Address offset: 0x570

End point of 5th piece wise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000050				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	T4			End point of 5th piece wise linear function

6.28.2 Electrical specification

6.28.2.1 Temperature Sensor Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{TEMP}	Time required for temperature measurement		36		µs
$T_{TEMP,RANGE}$	Temperature sensor range	-40		85	°C
$T_{TEMP,ACC}$	Temperature sensor accuracy	-5		5	°C
$T_{TEMP,RES}$	Temperature sensor resolution		0.25		°C
$T_{TEMP,STB}$	Sample to sample stability at constant device temperature	-0.25		0.25	°C
$T_{TEMP,OFFST}$	Sample offset at 25°C	-2.5		2.5	°C

6.29 TWI — I^2C compatible two-wire interface

The TWI master is compatible with I^2C operating at 100 kHz and 400 kHz.

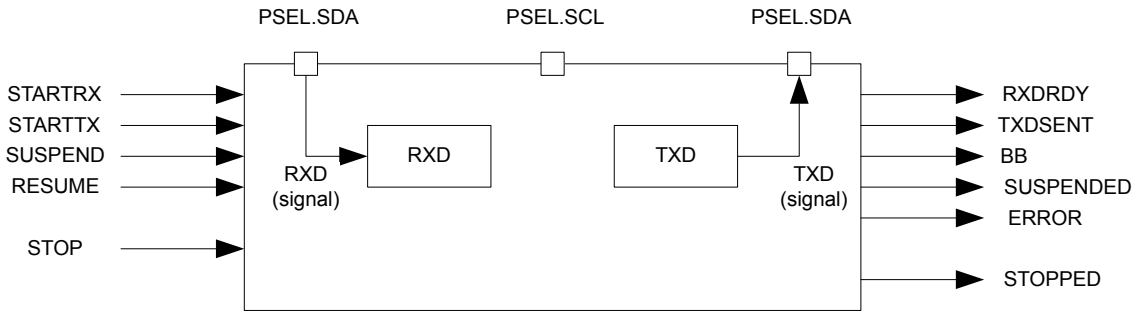

Figure 165: TWI master's main features

6.29.1 Functional description

This TWI master is not compatible with CBUS. The TWI transmitter and receiver are single buffered.

See, TWI master's main features on page 411.

A TWI setup comprising one master and three slaves is illustrated in A typical TWI setup comprising one master and three slaves on page 411. This TWI master is only able to operate as the only master on the TWI bus.

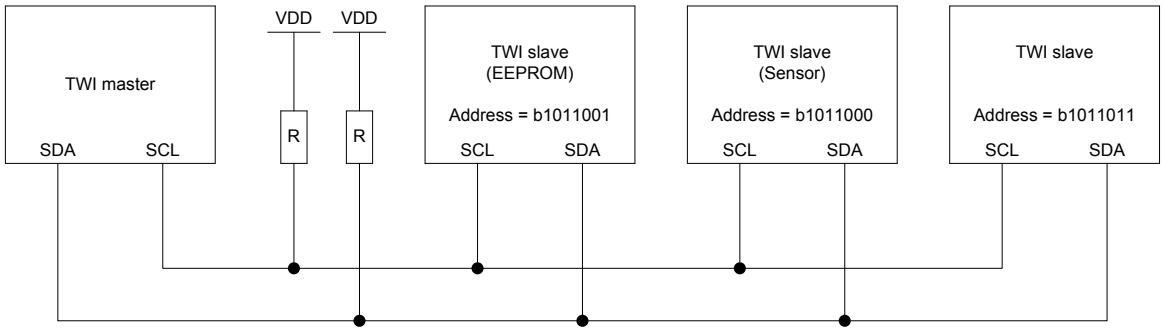

Figure 166: A typical TWI setup comprising one master and three slaves

This TWI master supports clock stretching performed by the slaves. The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

6.29.2 Master mode pin configuration

The different signals SCL and SDA associated with the TWI master are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

If the CONNECT field of a PSEL.xxx register is set to Disconnected, the associated TWI signal is not connected to any physical pin. The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI master is enabled, and retained only as long as the device is in ON mode. PSEL.SCL and PSEL.SDA must only be configured when the TWI is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration on page 412.

Only one peripheral can be assigned to drive a particular GPIO pin at a time, failing to do so may result in unpredictable behavior.

TWI master signal	TWI master pin	Direction	Drive strength	Output value
SCL	As specified in PSEL.SCL	Input	S0D1	Not applicable
SDA	As specified in PSEL.SDA	Input	S0D1	Not applicable

Table 112: GPIO configuration

6.29.3 Shared resources

The TWI shares registers and other resources with other peripherals that have the same ID as the TWI.

Therefore, you must disable all peripherals that have the same ID as the TWI before the TWI can be configured and used. Disabling a peripheral that has the same ID as the TWI will not reset any of the registers that are shared with the TWI. It is therefore important to configure all relevant TWI registers explicitly to secure that it operates correctly.

The Instantiation table in Instantiation on page 22 shows which peripherals have the same ID as the TWI.

6.29.4 Master write sequence

A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes that are written to the TXD register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave. A TXDSENT event will be generated each time the TWI master has clocked out a TXD byte, and the associated ACK/NACK bit has been clocked in from the slave.

The TWI master transmitter is single buffered, and a second byte can only be written to the TXD register after the previous byte has been clocked out and the ACK/NACK bit clocked in, that is, after the TXDSENT event has been generated.

If the CPU is prevented from writing to TXD when the TWI master is ready to clock out a byte, the TWI master will stretch the clock until the CPU has written a byte to the TXD register.

A typical TWI master write sequence is illustrated in The TWI master writing data to a slave on page 412. Occurrence 3 in the figure illustrates delayed processing of the TXDSENT event associated with TXD byte 1. In this scenario the TWI master will stretch the clock to prevent writing erroneous data to the slave.

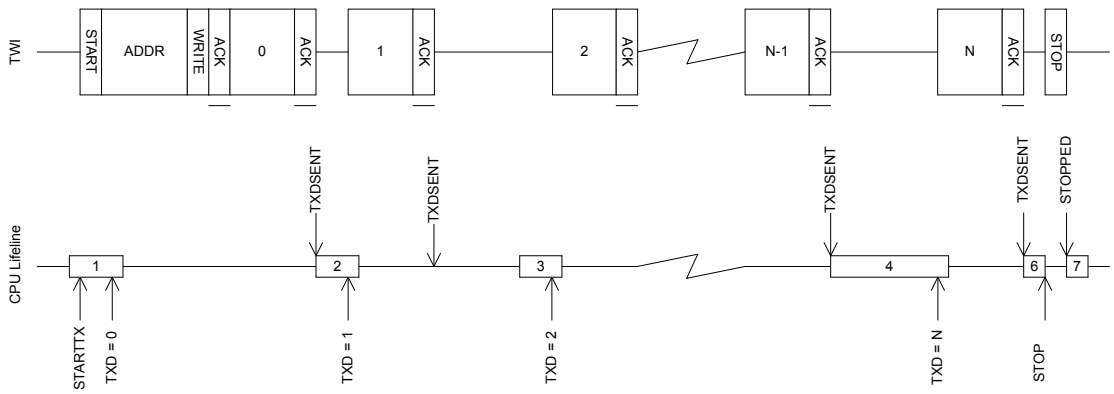

Figure 167: The TWI master writing data to a slave

The TWI master write sequence is stopped when the STOP task is triggered whereupon the TWI master will generate a stop condition on the TWI bus.

6.29.5 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1).

The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

The TWI master will generate a RXDRDY event every time a new byte is received in the RXD register.

After receiving a byte, the TWI master will delay sending the ACK/NACK bit by stretching the clock until the CPU has extracted the received byte, that is, by reading the RXD register.

The TWI master read sequence is stopped by triggering the STOP task. This task must be triggered before the last byte is extracted from RXD to ensure that the TWI master sends a NACK back to the slave before generating the stop condition.

A typical TWI master read sequence is illustrated in The TWI master reading data from a slave on page 413. Occurrence 3 in this figure illustrates delayed processing of the RXDRDY event associated with RXD byte B. In this scenario the TWI master will stretch the clock to prevent the slave from overwriting the contents of the RXD register.

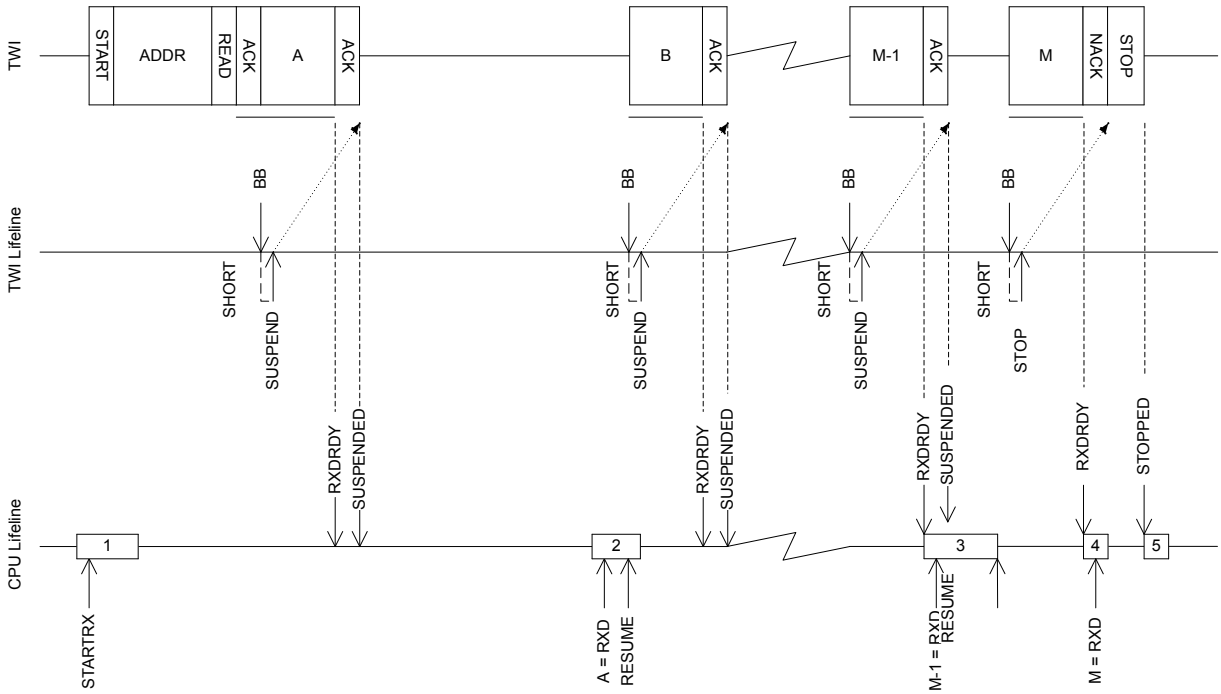

Figure 168: The TWI master reading data from a slave

6.29.6 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes one byte to the slave followed by reading M bytes from the slave. Any combination and number of transmit and receive sequences can be combined in this fashion. Only one shortcut to STOP can be enabled at any given time.

The figure below illustrates a repeated start sequence where the TWI master writes one byte, followed by reading M bytes from the slave without performing a stop in-between.

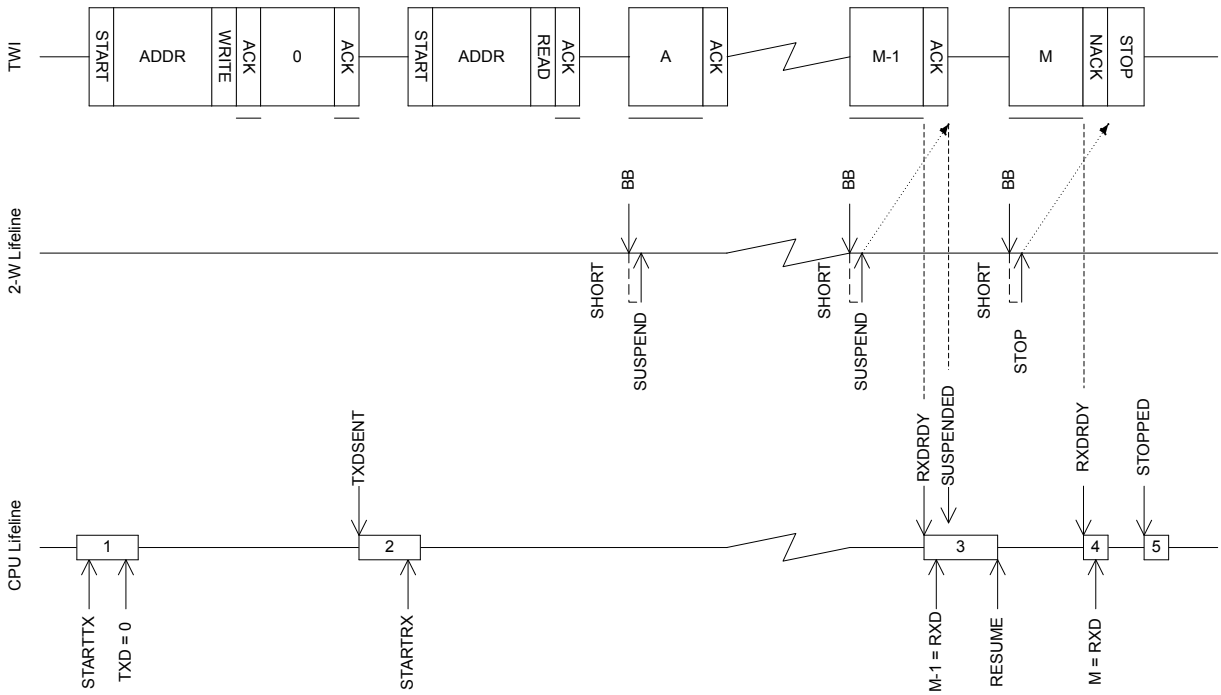

Figure 169: A repeated start sequence, where the TWI master writes one byte, followed by reading M bytes from the slave without performing a stop in-between

To generate a repeated start after a read sequence, a second start task must be triggered instead of the STOP task, that is, STARTRX or STARTTX. This start task must be triggered before the last byte is extracted from RXD to ensure that the TWI master sends a NACK back to the slave before generating the repeated start condition.

6.29.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.29.8 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWI	TWI0	Two-wire interface master 0	Deprecated
0x40004000	TWI	TWI1	Two-wire interface master 1	Deprecated

Table 113: Instances

Register	Offset	Description
TASKS_STARTRX	0x000	Start TWI receive sequence
TASKS_STARTTX	0x008	Start TWI transmit sequence
TASKS_STOP	0x014	Stop TWI transaction
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction

Register	Offset	Description
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_RXDREADY	0x108	TWI RXD byte received
EVENTS_TXDSENT	0x11C	TWI TXD byte sent
EVENTS_ERROR	0x124	TWI error
EVENTS_BB	0x138	TWI byte boundary, generated before each byte that is sent or received
EVENTS_SUSPENDED	0x148	TWI entered the suspended state
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x4C4	Error source
ENABLE	0x500	Enable TWI
PSEL.SCL	0x508	Pin select for SCL
PSEL.SDA	0x50C	Pin select for SDA
RXD	0x518	RXD register
TXD	0x51C	TXD register
FREQUENCY	0x524	TWI frequency. Accuracy depends on the HFCLK source selected.
ADDRESS	0x588	Address used in the TWI transfer

Table 114: Register overview

6.29.8.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	BB_SUSPEND			Shortcut between BB event and SUSPEND task See EVENTS_BB and TASKS_SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	BB_STOP			Shortcut between BB event and STOP task See EVENTS_BB and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.29.8.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID														F				E				D		C					B	A		
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
B	RW	RXDREADY			Write '1' to enable interrupt for RXDREADY event See EVENTS_RXDREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	TXDSENT			Write '1' to enable interrupt for TXDSENT event See EVENTS_TXDSENT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	BB			Write '1' to enable interrupt for BB event See EVENTS_BB
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to enable interrupt for SUSPENDED event See EVENTS_SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.29.8.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	RXDREADY			Write '1' to disable interrupt for RXDREADY event See EVENTS_RXDREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	TXDSENT			Write '1' to disable interrupt for TXDSENT event See EVENTS_TXDSENT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	BB			Write '1' to disable interrupt for BB event See EVENTS_BB
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to disable interrupt for SUSPENDED event See EVENTS_SUSPENDED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.29.8.4 ERRORSRC

Address offset: 0x4C4

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A new byte was received before previous byte got read by software from the RXD register. (Previous data is lost)
			NotPresent	0	Read: no overrun occured
			Present	1	Read: overrun occured
B	RW	ANACK			NACK received after sending the address (write '1' to clear)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
C	RW	DNACK			NACK received after sending a data byte (write '1' to clear)
			NotPresent	0	Read: error not present
			Present	1	Read: error present

6.29.8.5 ENABLE

Address offset: 0x500

Enable TWI

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWI
			Disabled	0	Disable TWI
			Enabled	5	Enable TWI

6.29.8.6 PSEL.SCL

Address offset: 0x508

Pin select for SCL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.29.8.7 PSEL.SDA

Address offset: 0x50C

Pin select for SDA

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.29.8.8 RXD

Address offset: 0x518

RXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXD			RXD register

6.29.8.9 TXD

Address offset: 0x51C

TXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TXD			TXD register

6.29.8.10 FREQUENCY

Address offset: 0x524

TWI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQUENCY			TWI master clock frequency
			K100	0x01980000	100 kbps
			K250	0x04000000	250 kbps
			K400	0x06680000	400 kbps (actual rate 410.256 kbps)

6.29.8.11 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

6.29.9 Electrical specification

6.29.9.1 TWI interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWI,SCL}$	Bit rates for TWI[37]	100		400	kbps
$t_{TWI,START}$	Time from STARTRX/STARTTX task to transmission started		1.5		µs

6.29.9.2 Two Wire Interface (TWI) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWI,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWI,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns

[37] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TWI,HD\_STA,100kbps}$	TWI master hold time for START and repeated START condition, 100 kbps	10000			ns
$t_{TWI,HD\_STA,250kbps}$	TWI master hold time for START and repeated START condition, 250kbps	4000			ns
$t_{TWI,HD\_STA,400kbps}$	TWI master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWI,SU\_STO,100kbps}$	TWI master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWI,SU\_STO,250kbps}$	TWI master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWI,SU\_STO,400kbps}$	TWI master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWI,BUF,100kbps}$	TWI master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWI,BUF,250kbps}$	TWI master bus free time between STOP and START conditions, 250 kbps	2700			ns
$t_{TWI,BUF,400kbps}$	TWI master bus free time between STOP and START conditions, 400 kbps	2100			ns

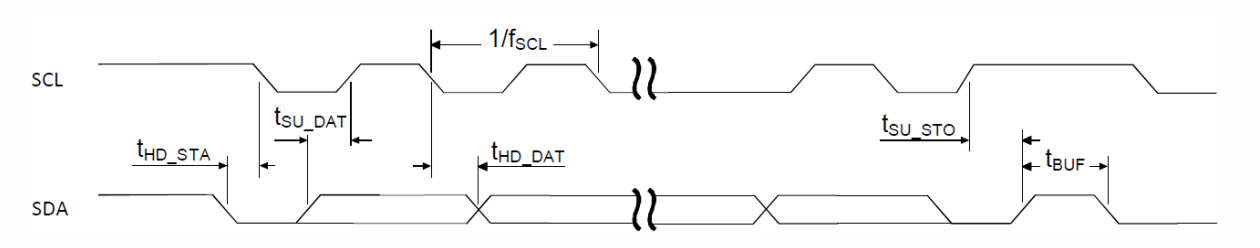

Figure 170: TWI timing diagram, 1 byte transaction

6.30 TIMER — Timer/counter

The TIMER can operate in two modes: timer and counter.

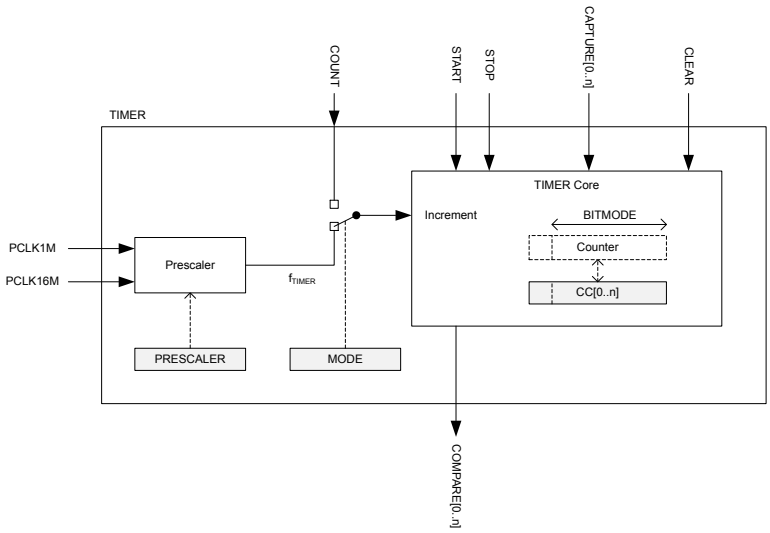

Figure 171: Block schematic for timer/counter

The timer/counter runs on the high-frequency clock source (HFCLK) and includes a four-bit (1/2X) prescaler that can divide the timer input clock from the HFCLK controller. Clock source selection between PCLK16M and PCLK1M is automatic according to TIMER base frequency set by the prescaler. The TIMER base frequency is always given as 16 MHz divided by the prescaler value.

The PPI system allows a TIMER event to trigger a task of any other system peripheral of the device. The PPI system also enables the TIMER task/event features to generate periodic output and PWM signals to any GPIO. The number of input/outputs used at the same time is limited by the number of GPIOTE channels.

The TIMER can operate in two modes, Timer mode and Counter mode. In both modes, the TIMER is started by triggering the START task, and stopped by triggering the STOP task. After the timer is stopped the timer can resume timing/counting by triggering the START task again. When timing/counting is resumed, the timer will continue from the value it had prior to being stopped.

In Timer mode, the TIMER's internal Counter register is incremented by one for every tick of the timer frequency f_{TIMER} as illustrated in Block schematic for timer/counter on page 420. The timer frequency is derived from PCLK16M as shown below, using the values specified in the PRESCALER register:

$$f_{TIMER} = 16\ MHz\ /\ (2^{PRESCALER})$$

When f_{TIMER} <= 1 MHz the TIMER will use PCLK1M instead of PCLK16M for reduced power consumption.

In counter mode, the TIMER's internal Counter register is incremented by one each time the COUNT task is triggered, that is, the timer frequency and the prescaler are not utilized in counter mode. Similarly, the COUNT task has no effect in Timer mode.

The TIMER's maximum value is configured by changing the bit-width of the timer in the BITMODE on page 425 register.

PRESCALER on page 425 and the BITMODE on page 425 must only be updated when the timer is stopped. If these registers are updated while the TIMER is started then this may result in unpredictable behavior.

When the timer is incremented beyond its maximum value the Counter register will overflow and the TIMER will automatically start over from zero.

The Counter register can be cleared, that is, its internal value set to zero explicitly, by triggering the CLEAR task.

The TIMER implements multiple capture/compare registers.

Independent of prescaler setting the accuracy of the TIMER is equivalent to one tick of the timer frequency f_{TIMER} as illustrated in Block schematic for timer/counter on page 420.

6.30.1 Capture

The TIMER implements one capture task for every available capture/compare register.

Every time the CAPTURE[n] task is triggered, the Counter value is copied to the CC[n] register.

6.30.2 Compare

The TIMER implements one COMPARE event for every available capture/compare register.

A COMPARE event is generated when the Counter is incremented and then becomes equal to the value specified in one of the capture compare registers. When the Counter value becomes equal to the value specified in a capture compare register CC[n], the corresponding compare event COMPARE[n] is generated.

BITMODE on page 425 specifies how many bits of the Counter register and the capture/compare register that are used when the comparison is performed. Other bits will be ignored.

6.30.3 Task delays

After the TIMER is started, the CLEAR task, COUNT task and the STOP task will guarantee to take effect within one clock cycle of the PCLK16M.

6.30.4 Task priority

If the START task and the STOP task are triggered at the same time, that is, within the same period of PCLK16M, the STOP task will be prioritized.

6.30.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40008000	TIMER	TIMER0	Timer 0	This timer instance has 4 CC registers (CC[0..3])
0x40009000	TIMER	TIMER1	Timer 1	This timer instance has 4 CC registers (CC[0..3])
0x4000A000	TIMER	TIMER2	Timer 2	This timer instance has 4 CC registers (CC[0..3])
0x4001A000	TIMER	TIMER3	Timer 3	This timer instance has 6 CC registers (CC[0..5])
0x4001B000	TIMER	TIMER4	Timer 4	This timer instance has 6 CC registers (CC[0..5])

Table 115: Instances

Register	Offset	Description	
TASKS_START	0x000	Start Timer	
TASKS_STOP	0x004	Stop Timer	
TASKS_COUNT	0x008	Increment Timer (Counter mode only)	
TASKS_CLEAR	0x00C	Clear time	
TASKS_SHUTDOWN	0x010	Shut down timer	Deprecated
TASKS_CAPTURE[0]	0x040	Capture Timer value to CC[0] register	
TASKS_CAPTURE[1]	0x044	Capture Timer value to CC[1] register	
TASKS_CAPTURE[2]	0x048	Capture Timer value to CC[2] register	
TASKS_CAPTURE[3]	0x04C	Capture Timer value to CC[3] register	
TASKS_CAPTURE[4]	0x050	Capture Timer value to CC[4] register	
TASKS_CAPTURE[5]	0x054	Capture Timer value to CC[5] register	
EVENTS_COMPARE[0]	0x140	Compare event on CC[0] match	
EVENTS_COMPARE[1]	0x144	Compare event on CC[1] match	
EVENTS_COMPARE[2]	0x148	Compare event on CC[2] match	
EVENTS_COMPARE[3]	0x14C	Compare event on CC[3] match	
EVENTS_COMPARE[4]	0x150	Compare event on CC[4] match	
EVENTS_COMPARE[5]	0x154	Compare event on CC[5] match	
SHORTS	0x200	Shortcut register	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
MODE	0x504	Timer mode selection	
BITMODE	0x508	Configure the number of bits used by the TIMER	
PRESCALER	0x510	Timer prescaler register	
CC[0]	0x540	Capture/Compare register 0	

Register	Offset	Description
CC[1]	0x544	Capture/Compare register 1
CC[2]	0x548	Capture/Compare register 2
CC[3]	0x54C	Capture/Compare register 3
CC[4]	0x550	Capture/Compare register 4
CC[5]	0x554	Capture/Compare register 5

Table 116: Register overview

6.30.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-F	RW	COMPARE[i]_CLEAR (i=0..5)			Shortcut between COMPARE[i] event and CLEAR task See EVENTS_COMPARE[i] and TASKS_CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G-L	RW	COMPARE[i]_STOP (i=0..5)			Shortcut between COMPARE[i] event and STOP task See EVENTS_COMPARE[i] and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.30.5.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	COMPARE0			Write '1' to enable interrupt for COMPARE[0] event See EVENTS_COMPARE[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	COMPARE1			Write '1' to enable interrupt for COMPARE[1] event See EVENTS_COMPARE[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE2			Write '1' to enable interrupt for COMPARE[2] event See EVENTS_COMPARE[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
D	RW	COMPARE3			Write '1' to enable interrupt for COMPARE[3] event See EVENTS_COMPARE[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE4			Write '1' to enable interrupt for COMPARE[4] event See EVENTS_COMPARE[4]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE5			Write '1' to enable interrupt for COMPARE[5] event See EVENTS_COMPARE[5]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.30.5.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	COMPARE0			Write '1' to disable interrupt for COMPARE[0] event See EVENTS_COMPARE[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	COMPARE1			Write '1' to disable interrupt for COMPARE[1] event See EVENTS_COMPARE[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE2			Write '1' to disable interrupt for COMPARE[2] event See EVENTS_COMPARE[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	COMPARE3			Write '1' to disable interrupt for COMPARE[3] event See EVENTS_COMPARE[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE4			Write '1' to disable interrupt for COMPARE[4] event See EVENTS_COMPARE[4]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE5			Write '1' to disable interrupt for COMPARE[5] event See EVENTS_COMPARE[5]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.30.5.4 MODE

Address offset: 0x504

Timer mode selection

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MODE			Timer mode
			Timer	0	Select Timer mode
			Counter	1	Select Counter mode
			LowPowerCounter	2	Select Low Power Counter mode

Deprecated

6.30.5.5 BITMODE

Address offset: 0x508

Configure the number of bits used by the TIMER

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	BITMODE			Timer bit width
			16Bit	0	16 bit timer bit width
			08Bit	1	8 bit timer bit width
			24Bit	2	24 bit timer bit width
			32Bit	3	32 bit timer bit width

6.30.5.6 PRESCALER

Address offset: 0x510

Timer prescaler register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A
Reset 0x00000004	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PRESCALER		[0..9]	Prescaler value

6.30.5.7 CC[n] (n=0..5)

Address offset: 0x540 + (n × 0x4)

Capture/Compare register n

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CC			Capture/Compare value Only the number of bits indicated by BITMODE will be used by the TIMER.

6.31 TWIM — I²C compatible two-wire interface master with EasyDMA

TWI master with EasyDMA (TWIM) is a two-wire half-duplex master which can communicate with multiple slave devices connected to the same bus

Listed here are the main features for TWIM:

- I²C compatible
- 100 kbps, 250 kbps, or 400 kbps
- Support for clock stretching (non I²C compliant)
- EasyDMA

The two-wire interface can communicate with a bi-directional wired-AND bus with two lines (SCL, SDA). The protocol makes it possible to interconnect up to 127 individually addressable devices. TWIM is not compatible with CBUS.

The GPIOs used for each two-wire interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

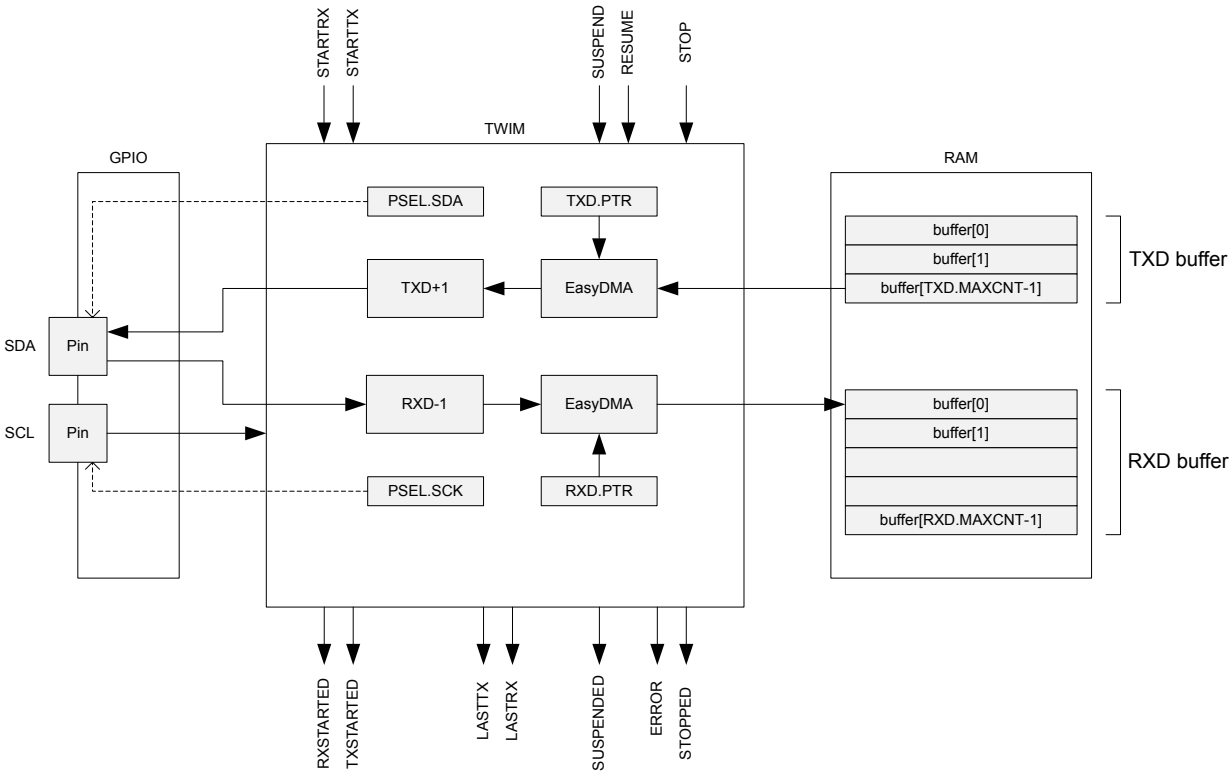

Figure 172: TWI master with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 427. This TWIM is only able to operate as a single master on the TWI bus. Multi-master bus configuration is not supported.

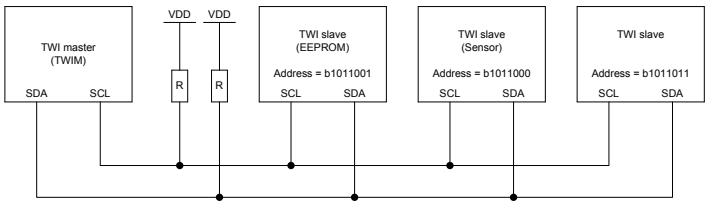

Figure 173: A typical TWI setup comprising one master and three slaves

This TWI master supports clock stretching performed by the slaves. Note that the SCK pulse following a stretched clock cycle may be shorter than specified by the I2C specification.

The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task. The TWI master will generate a STOPPED event when it has stopped following a STOP task. The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

After the TWI master is started, the STARTTX task or the STARTRX task should not be triggered again before the TWI master has stopped, i.e. following a LASTRX, LASTTX or STOPPED event.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

6.31.1 EasyDMA

The TWI master implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

6.31.2 Master write sequence

A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes found in the transmit buffer located in RAM at the address specified in the TXD.PTR register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave.

A typical TWI master write sequence is illustrated in TWI master writing data to a slave on page 428. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

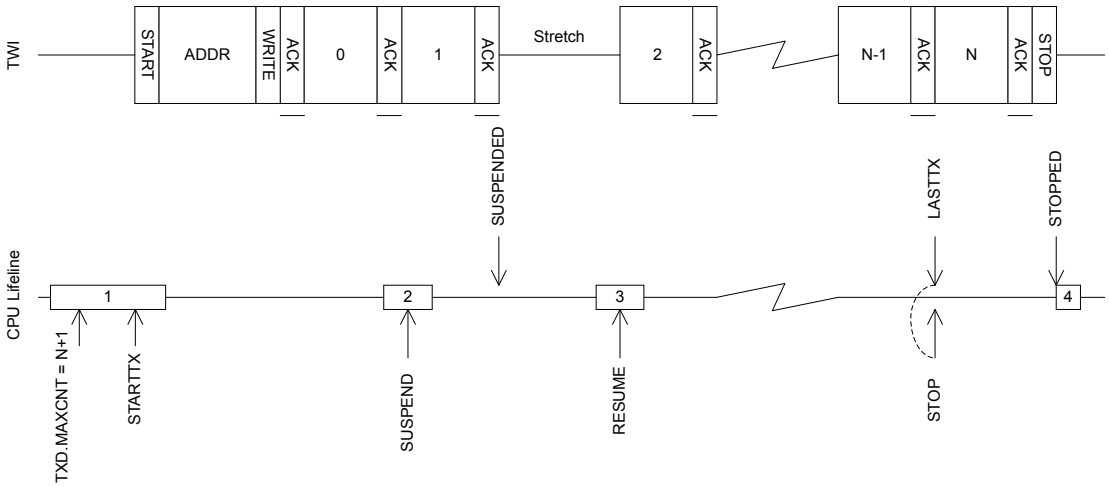

Figure 174: TWI master writing data to a slave

The TWI master will generate a LASTTX event when it starts to transmit the last byte, this is illustrated in TWI master writing data to a slave on page 428

The TWI master is stopped by triggering the STOP task, this task should be triggered during the transmission of the last byte to secure that the TWI will stop as fast as possible after sending the last byte. It is safe to use the shortcut between LASTTX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the whole RAM buffer has been sent, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

6.31.3 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the

address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1). The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

Data received will be stored in RAM at the address specified in the RXD.PTR register. The TWI master will generate an ACK after all but the last byte received from the slave. The TWI master will generate a NACK after the last byte received to indicate that the read sequence shall stop.

A typical TWI master read sequence is illustrated in The TWI master reading data from a slave on page 429. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

The TWI master will generate a LASTRX event when it is ready to receive the last byte, this is illustrated in The TWI master reading data from a slave on page 429. If RXD.MAXCNT > 1 the LASTRX event is generated after sending the ACK of the previously received byte. If RXD.MAXCNT = 1 the LASTRX event is generated after receiving the ACK following the address and READ bit.

The TWI master is stopped by triggering the STOP task, this task must be triggered before the NACK bit is supposed to be transmitted. The STOP task can be triggered at any time during the reception of the last byte. It is safe to use the shortcut between LASTRX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the RAM buffer is full, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

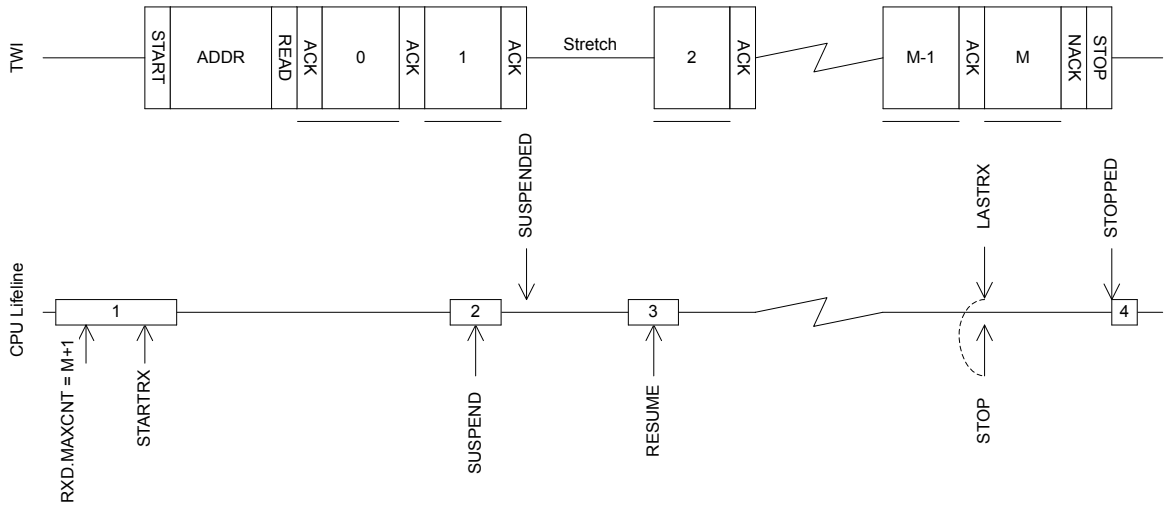

Figure 175: The TWI master reading data from a slave

6.31.4 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave. This example uses shortcuts to perform the simplest type of repeated start sequence, i.e. one write followed by one read. The same approach can be used to perform a repeated start sequence where the sequence is read followed by write.

The figure A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave on page 430 illustrates this:

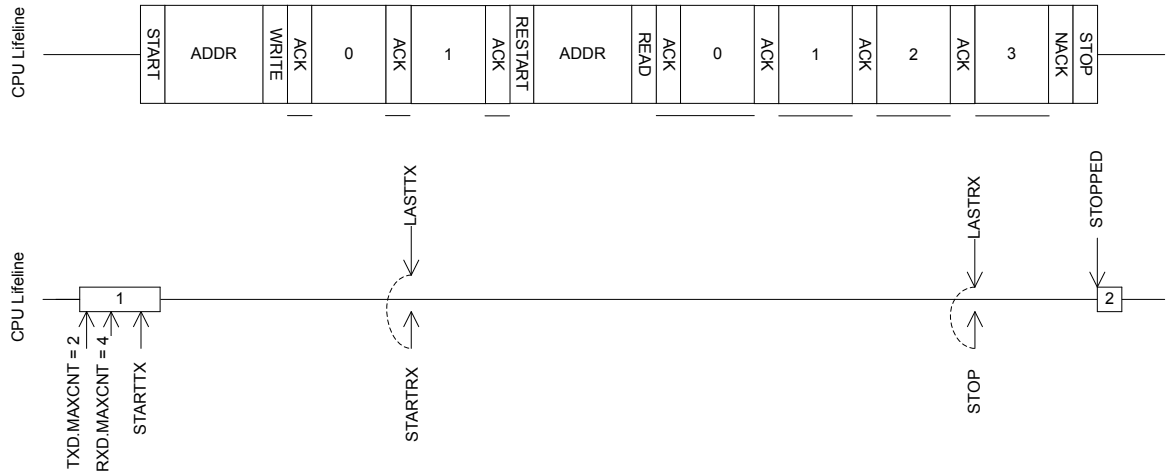

Figure 176: A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave

If a more complex repeated start sequence is needed and the TWI firmware drive is serviced in a low priority interrupt it may be necessary to use the SUSPEND task and SUSPENDED event to guarantee that the correct tasks are generated at the correct time. This is illustrated in A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts on page 430.

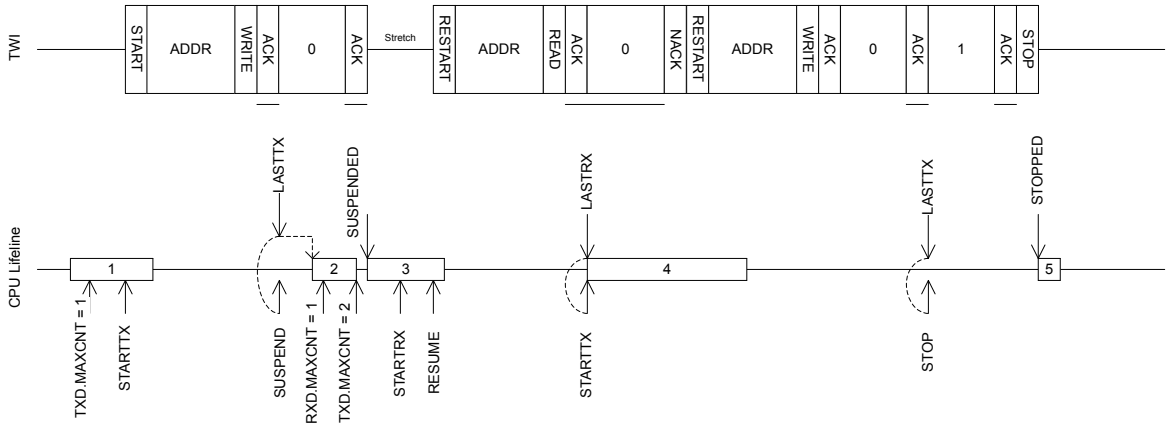

Figure 177: A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts

6.31.5 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.31.6 Master mode pin configuration

The SCL and SDA signals associated with the TWI master are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI master is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins

will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL, PSEL.SDA must only be configured when the TWI master is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 431.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI master signal	TWI master pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 117: GPIO configuration before enabling peripheral

6.31.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWIM	TWIM0	Two-wire interface master 0	
0x40004000	TWIM	TWIM1	Two-wire interface master 1	

Table 118: Instances

Register	Offset	Description
TASKS_STARTRX	0x000	Start TWI receive sequence
TASKS_STARTTX	0x008	Start TWI transmit sequence
TASKS_STOP	0x014	Stop TWI transaction. Must be issued while the TWI master is not suspended.
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x124	TWI error
EVENTS_SUSPENDED	0x148	Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
EVENTS_RXSTARTED	0x14C	Receive sequence started
EVENTS_TXSTARTED	0x150	Transmit sequence started
EVENTS_LASTRX	0x15C	Byte boundary, starting to receive the last byte
EVENTS_LASTTX	0x160	Byte boundary, starting to transmit the last byte
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x4C4	Error source
ENABLE	0x500	Enable TWIM
PSEL.SCL	0x508	Pin select for SCL signal
PSEL.SDA	0x50C	Pin select for SDA signal
FREQUENCY	0x524	TWI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534	Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last transaction
RXD.LIST	0x540	EasyDMA list type
TXD.PTR	0x544	Data pointer
TXD.MAXCNT	0x548	Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C	Number of bytes transferred in the last transaction

Register	Offset	Description
TXD.LIST	0x550	EasyDMA list type
ADDRESS	0x588	Address used in the TWI transfer

Table 119: Register overview

6.31.7.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	LASTTX_STARTRX			Shortcut between LASTTX event and STARTRX task See EVENTS_LASTTX and TASKS_STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	LASTTX_SUSPEND			Shortcut between LASTTX event and SUSPEND task See EVENTS_LASTTX and TASKS_SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LASTTX_STOP			Shortcut between LASTTX event and STOP task See EVENTS_LASTTX and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LASTRX_STARTTX			Shortcut between LASTRX event and STARTTX task See EVENTS_LASTRX and TASKS_STARTTX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LASTRX_SUSPEND			Shortcut between LASTRX event and SUSPEND task See EVENTS_LASTRX and TASKS_SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	LASTRX_STOP			Shortcut between LASTRX event and STOP task See EVENTS_LASTRX and TASKS_STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.31.7.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	J I H G F D A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	J I H G F D A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ERROR			Enable or disable interrupt for ERROR event See EVENTS_ERROR
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	SUSPENDED			Enable or disable interrupt for SUSPENDED event See EVENTS_SUSPENDED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXSTARTED			Enable or disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	TXSTARTED			Enable or disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	LASTRX			Enable or disable interrupt for LASTRX event See EVENTS_LASTRX
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	LASTTX			Enable or disable interrupt for LASTTX event See EVENTS_LASTTX
			Disabled	0	Disable
			Enabled	1	Enable

6.31.7.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	J I H G F D A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
F	RW	SUSPENDED			Write '1' to enable interrupt for SUSPENDED event See EVENTS_SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to enable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to enable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to enable interrupt for LASTRX event See EVENTS_LASTRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to enable interrupt for LASTTX event See EVENTS_LASTTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.31.7.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to disable interrupt for SUSPENDED event See EVENTS_SUSPENDED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to disable interrupt for LASTRX event See EVENTS_LASTRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to disable interrupt for LASTTX event See EVENTS_LASTTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.31.7.5 ERRORSRC

Address offset: 0x4C4

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A new byte was received before previous byte got transferred into RXD buffer. (Previous data is lost)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
B	RW	ANACK			NACK received after sending the address (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	DNACK			NACK received after sending a data byte (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred

6.31.7.6 ENABLE

Address offset: 0x500

Enable TWIM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIM
			Disabled	0	Disable TWIM
			Enabled	6	Enable TWIM

6.31.7.7 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.31.7.8 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.31.7.9 FREQUENCY

Address offset: 0x524

TWI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	FREQUENCY			TWI master clock frequency
			K100	0x01980000	100 kbps
			K250	0x04000000	250 kbps
			K400	0x06400000	400 kbps

6.31.7.10 RXD.PTR

Address offset: 0x534

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.31.7.11 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

6.31.7.12 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.31.7.13 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.31.7.14 TXD.PTR

Address offset: 0x544

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.31.7.15 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

6.31.7.16 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.31.7.17 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.31.7.18 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

6.31.8 Electrical specification

6.31.8.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL}$	Bit rates for TWIM[38]	100		400	kbps
$t_{TWIM,START}$	Time from STARTRX/STARTTX task to transmission started		1.5		µs

6.31.8.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWIM,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIM,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIM,HD\_STA,100kbps}$	TWIM master hold time for START and repeated START condition, 100 kbps	10000			ns
$t_{TWIM,HD\_STA,250kbps}$	TWIM master hold time for START and repeated START condition, 250kbps	4000			ns
$t_{TWIM,HD\_STA,400kbps}$	TWIM master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWIM,SU\_STO,100kbps}$	TWIM master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWIM,SU\_STO,250kbps}$	TWIM master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWIM,SU\_STO,400kbps}$	TWIM master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWIM,BUF,100kbps}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWIM,BUF,250kbps}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2700			ns

[38] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TWIM,BUF,400kbps}$	TWIM master bus free time between STOP and START conditions, 400 kbps	2100			ns

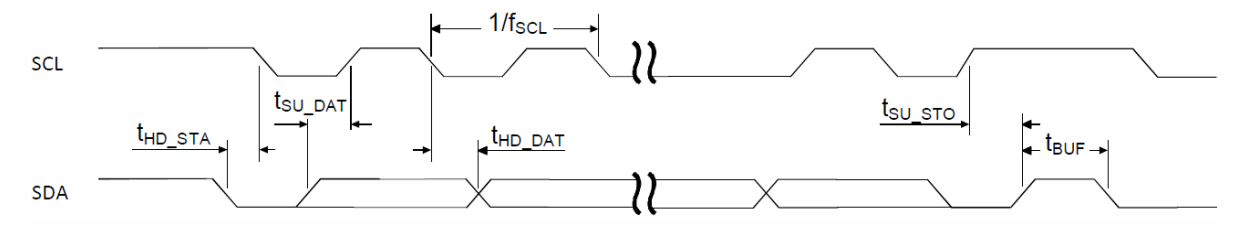

Figure 178: TWIM timing diagram, 1 byte transaction

6.31.9 Pullup resistor

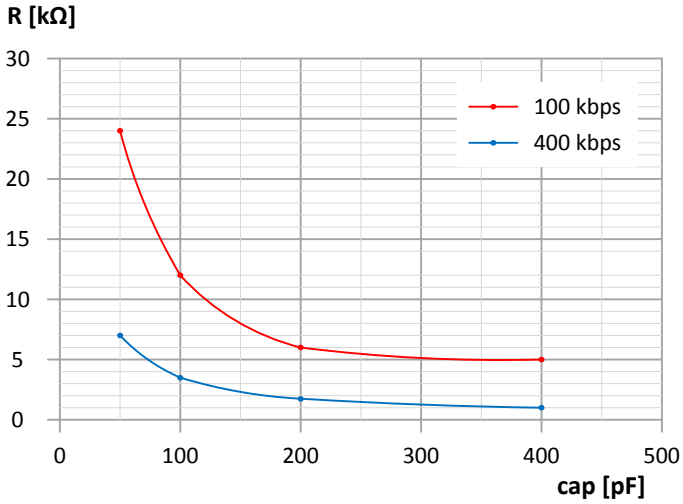

Figure 179: Recommended TWIM pullup value vs. line capacitance

- The I2C specification allows a line capacitance of 400 pF at most.
- The value of internal pullup resistor (R_{PU}) for nRF52840 can be found in GPIO — General purpose input/output on page 141.

6.32 TWIS — I²C compatible two-wire interface slave with EasyDMA

TWI slave with EasyDMA (TWIS) is compatible with I²C operating at 100 kHz and 400 kHz. The TWI transmitter and receiver implement EasyDMA.

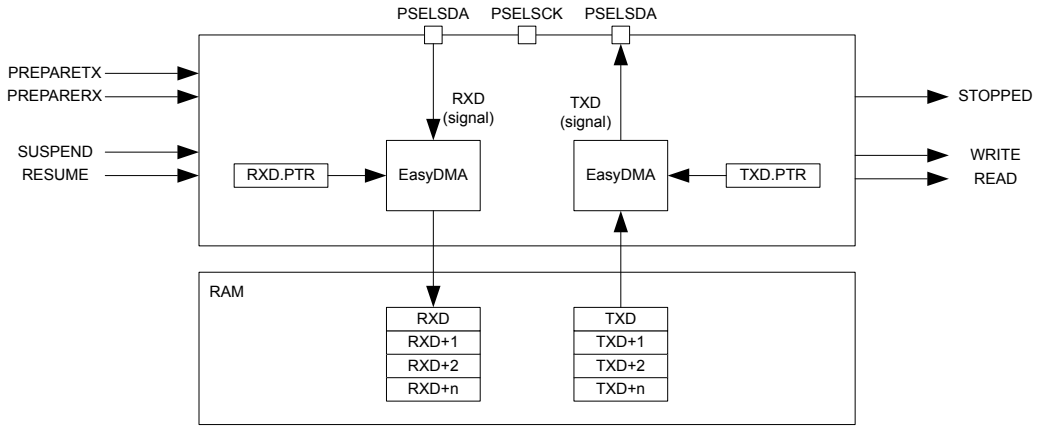

Figure 180: TWI slave with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 441. TWIS is only able to operate with a single master on the TWI bus.

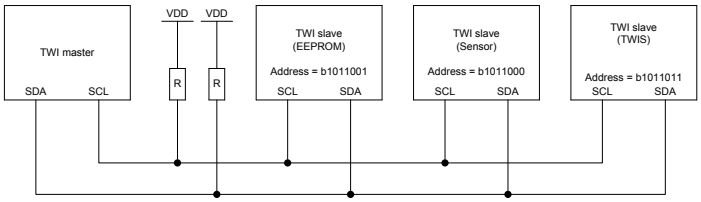

Figure 181: A typical TWI setup comprising one master and three slaves

The TWI slave state machine is illustrated in TWI slave state machine on page 442 and TWI slave state machine symbols on page 442 is explaining the different symbols used in the state machine.

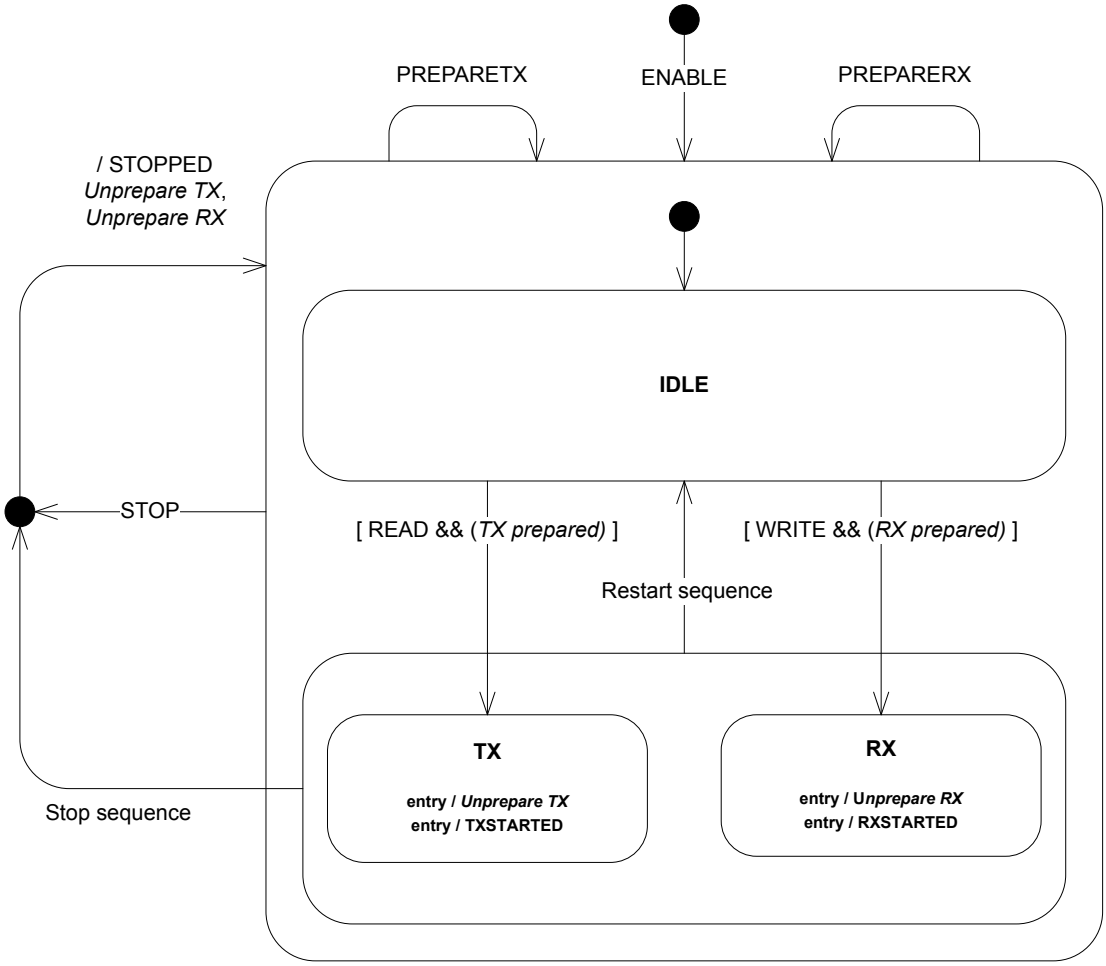

Figure 182: TWI slave state machine

Symbol	Type	Description
ENABLE	Register	The TWI slave has been enabled via the ENABLE register
PREPARETX	Task	The TASKS_PREPARETX task has been triggered
STOP	Task	The TASKS_STOP task has been triggered
PREPARERX	Task	The TASKS_PREPARERX task has been triggered
STOPPED	Event	The EVENTS_STOPPED event was generated
RXSTARTED	Event	The EVENTS_RXSTARTED event was generated
TXSTARTED	Event	The EVENTS_TXSTARTED event was generated
TX prepared	Internal	Internal flag indicating that a TASKS_PREPARETX task has been triggered. This flag is not visible to the user.
RX prepared	Internal	Internal flag indicating that a TASKS_PREPARERX task has been triggered. This flag is not visible to the user.
Unprepare TX	Internal	Clears the internal 'TX prepared' flag until next TASKS_PREPARETX task.
Unprepare RX	Internal	Clears the internal 'RX prepared' flag until next TASKS_PREPARERX task.
Stop sequence	TWI protocol	A TWI stop sequence was detected
Restart sequence	TWI protocol	A TWI restart sequence was detected

Table 120: TWI slave state machine symbols

The TWI slave supports clock stretching performed by the master.

The TWI slave operates in a low power mode while waiting for a TWI master to initiate a transfer. As long as the TWI slave is not addressed, it will remain in this low power mode.

To secure correct behaviour of the TWI slave, PSEL.SCL, PSEL.SDA, CONFIG and the ADDRESS[n] registers, must be configured prior to enabling the TWI slave through the ENABLE register. Similarly, changing these settings must be performed while the TWI slave is disabled. Failing to do so may result in unpredictable behaviour.

6.32.1 EasyDMA

The TWI slave implements EasyDMA for reading and writing to and from the RAM.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

6.32.2 TWI slave responding to a read command

Before the TWI slave can respond to a read command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE} .

A read command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the TWI slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the read command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a READ event when it acknowledges the read command.

The TWI slave is only able to detect a read command from the IDLE state.

The TWI slave will set an internal 'TX prepared' flag when the PREPARETX task is triggered.

When the read command is received the TWI slave will enter the TX state if the internal 'TX prepared' flag is set.

If the internal 'TX prepared' flag is not set when the read command is received, the TWI slave will stretch the master's clock until the PREPARETX task is triggered and the internal 'TX prepared' flag is set.

The TWI slave will generate the TXSTARTED event and clear the 'TX prepared' flag ('unprepare TX') when it enters the TX state. In this state the TWI slave will send the data bytes found in the transmit buffer to the master using the master's clock. The TWI slave will consume I_{TX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the TX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the 'TX prepared' flag ('unprepare TX') and go back to the IDLE state when it has stopped.

The transmit buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to send TXD.MAXCNT bytes from the transmit buffer for each transaction. If the TWI master forces the slave to send more than TXD.MAXCNT bytes, the slave will send the byte specified in the ORC register to the master instead. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see TXD.PTR etc., are latched when the TXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the 'TX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 446.

Each byte sent from the slave will be followed by an ACK/NACK bit sent from the master. The TWI master will generate a NACK following the last byte that it wants to receive to tell the slave to release the bus so that the TWI master can generate the stop condition. The TXD.AMOUNT register can be queried after a transaction to see how many bytes were sent.

A typical TWI slave read command response is illustrated in The TWI slave responding to a read command on page 444. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

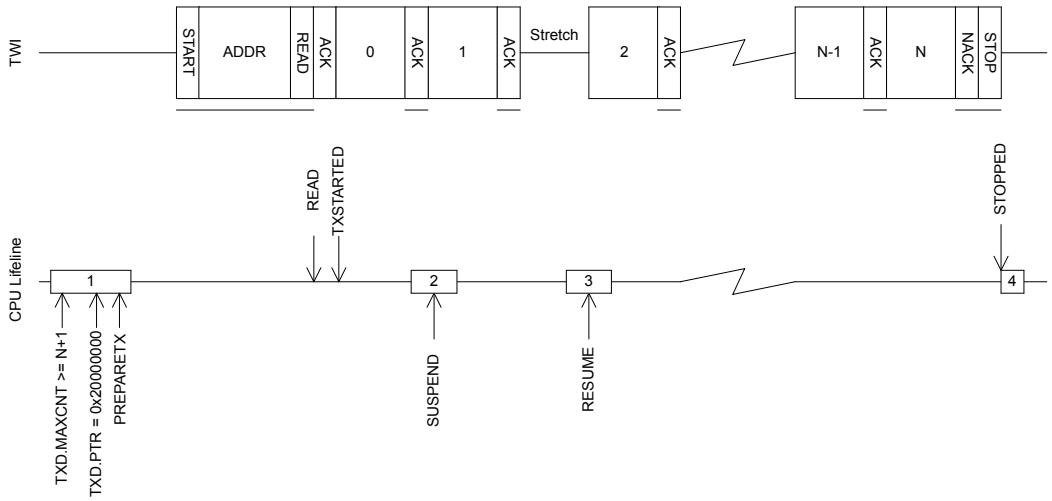

Figure 183: The TWI slave responding to a read command

6.32.3 TWI slave responding to a write command

Before the TWI slave can respond to a write command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE}.

A write command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the write command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a WRITE event if it acknowledges the write command.

The TWI slave is only able to detect a write command from the IDLE state.

The TWI slave will set an internal 'RX prepared' flag when the PREPARERX task is triggered.

When the write command is received the TWI slave will enter the RX state if the internal 'RX prepared' flag is set.

If the internal 'RX prepared' flag is not set when the write command is received, the TWI slave will stretch the master's clock until the PREPARERX task is triggered and the internal 'RX prepared' flag is set.

The TWI slave will generate the RXSTARTED event and clear the internal 'RX prepared' flag ('unprepare RX') when it enters the RX state. In this state the TWI slave will be able to receive the bytes sent by the TWI master. The TWI slave will consume I_{RX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the RX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the internal 'RX prepared' flag ('unprepare RX') and go back to the IDLE state when it has stopped.

The receive buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to receive as many bytes as specified in the RXD.MAXCNT register. If the TWI master tries to send more bytes to the slave than the slave is able to receive,these bytes will be discarded and the bytes will be NACKed by the slave. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see RXD.PTR etc., are latched when the RXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the internal 'RX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 446.

The TWI slave will generate an ACK after every byte received from the master. The RXD.AMOUNT register can be queried after a transaction to see how many bytes were received.

A typical TWI slave write command response is illustrated in The TWI slave responding to a write command on page 445. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

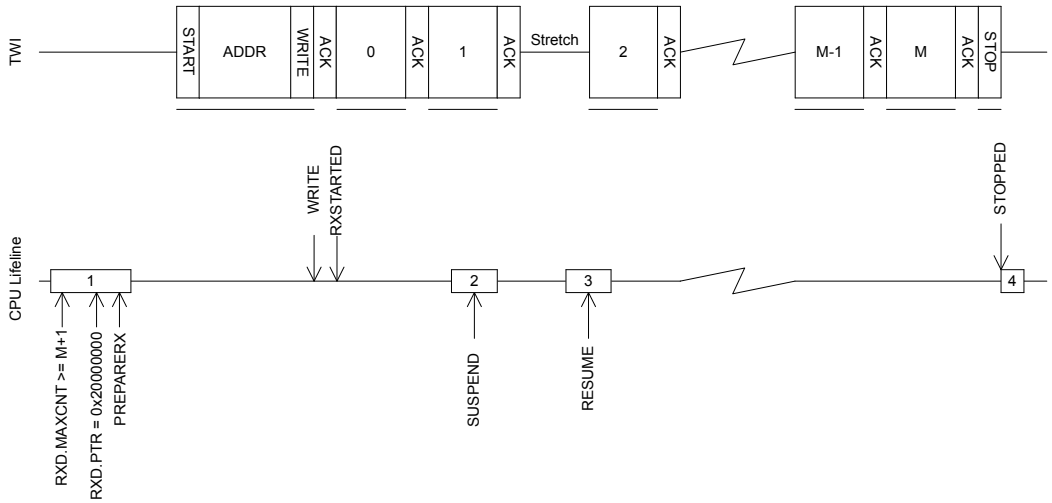

Figure 184: The TWI slave responding to a write command

6.32.4 Master repeated start sequence

An example of a repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave.

This is illustrated in A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave on page 446.

It is here assumed that the receiver does not know in advance what the master wants to read, and that this information is provided in the first two bytes received in the write part of the repeated start sequence. To guarantee that the CPU is able to process the received data before the TWI slave starts to reply to the read command, the SUSPEND task is triggered via a shortcut from the READ event generated when the read command is received. When the CPU has processed the incoming data and prepared the correct data response, the CPU will resume the transaction by triggering the RESUME task.

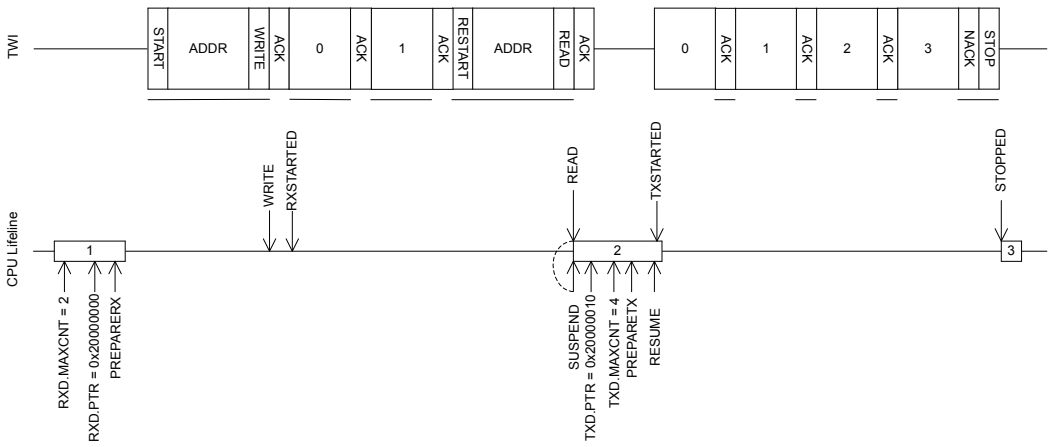

Figure 185: A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave

6.32.5 Terminating an ongoing TWI transaction

In some situations, e.g. if the external TWI master is not responding correctly, it may be required to terminate an ongoing transaction.

This can be achieved by triggering the STOP task. In this situation a STOPPED event will be generated when the TWI has stopped independent of whether or not a STOP condition has been generated on the TWI bus. The TWI slave will release the bus when it has stopped and go back to its IDLE state.

6.32.6 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.32.7 Slave mode pin configuration

The SCL and SDA signals associated with the TWI slave are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI slave is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL and PSEL.SDA must only be configured when the TWI slave is disabled.

To secure correct signal levels on the pins used by the TWI slave when the system is in OFF mode, and when the TWI slave is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 446.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI slave signal	TWI slave pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 121: GPIO configuration before enabling peripheral

6.32.8 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWIS	TWIS0	Two-wire interface slave 0	
0x40004000	TWIS	TWIS1	Two-wire interface slave 1	

Table 122: Instances

Register	Offset	Description
TASKS_STOP	0x014	Stop TWI transaction
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction
TASKS_PREPARERX	0x030	Prepare the TWI slave to respond to a write command
TASKS_PREPARETX	0x034	Prepare the TWI slave to respond to a read command
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x124	TWI error
EVENTS_RXSTARTED	0x14C	Receive sequence started
EVENTS_TXSTARTED	0x150	Transmit sequence started
EVENTS_WRITE	0x164	Write command received
EVENTS_READ	0x168	Read command received
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x4D0	Error source
MATCH	0x4D4	Status register indicating which address had a match
ENABLE	0x500	Enable TWIS
PSEL.SCL	0x508	Pin select for SCL signal
PSEL.SDA	0x50C	Pin select for SDA signal
RXD.PTR	0x534	RXD Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in RXD buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last RXD transaction
TXD.PTR	0x544	TXD Data pointer
TXD.MAXCNT	0x548	Maximum number of bytes in TXD buffer
TXD.AMOUNT	0x54C	Number of bytes transferred in the last TXD transaction
ADDRESS[0]	0x588	TWI slave address 0
ADDRESS[1]	0x58C	TWI slave address 1
CONFIG	0x594	Configuration register for the address match mechanism
ORC	0x5C0	Over-read character. Character sent out in case of an over-read of the transmit buffer.

Table 123: Register overview

6.32.8.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	WRITE_SUSPEND			Shortcut between WRITE event and SUSPEND task See EVENTS_WRITE and TASKS_SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	READ_SUSPEND			Shortcut between READ event and SUSPEND task See EVENTS_READ and TASKS_SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.32.8.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for STOPPED event See EVENTS_STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	ERROR			Enable or disable interrupt for ERROR event See EVENTS_ERROR
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	RXSTARTED			Enable or disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXSTARTED			Enable or disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	WRITE			Enable or disable interrupt for WRITE event See EVENTS_WRITE
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	READ			Enable or disable interrupt for READ event See EVENTS_READ
			Disabled	0	Disable
			Enabled	1	Enable

6.32.8.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for STOPPED event See EVENTS_STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to enable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to enable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to enable interrupt for WRITE event See EVENTS_WRITE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to enable interrupt for READ event See EVENTS_READ
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.32.8.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for STOPPED event See EVENTS_STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to disable interrupt for WRITE event See EVENTS_WRITE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to disable interrupt for READ event See EVENTS_READ
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.32.8.5 ERRORSRC

Address offset: 0x4D0

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERFLOW			RX buffer overflow detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred
B	RW	DNACK			NACK sent after receiving a data byte
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	OVERREAD			TX buffer over-read detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred

6.32.8.6 MATCH

Address offset: 0x4D4

Status register indicating which address had a match

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	MATCH		[0..1]	Which of the addresses in {ADDRESS} matched the incoming address

6.32.8.7 ENABLE

Address offset: 0x500

Enable TWIS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIS
			Disabled	0	Disable TWIS
			Enabled	9	Enable TWIS

6.32.8.8 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.32.8.9 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.32.8.10 RXD.PTR

Address offset: 0x534

RXD Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			RXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.32.8.11 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in RXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in RXD buffer

6.32.8.12 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last RXD transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last RXD transaction

6.32.8.13 TXD.PTR

Address offset: 0x544

TXD Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			TXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.32.8.14 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in TXD buffer

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in TXD buffer

6.32.8.15 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last TXD transaction

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last TXD transaction

6.32.8.16 ADDRESS[n] (n=0..1)

Address offset: 0x588 + (n × 0x4)

TWI slave address n

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ADDRESS			TWI slave address

6.32.8.17 CONFIG

Address offset: 0x594

Configuration register for the address match mechanism

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A
Reset 0x00000001					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A-B	RW	ADDRESS[i] (i=0..1)			Enable or disable address matching on ADDRESS[i]
			Disabled	0	Disabled
			Enabled	1	Enabled

6.32.8.18 ORC

Address offset: 0x5C0

Over-read character. Character sent out in case of an over-read of the transmit buffer.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character sent out in case of an over-read of the transmit buffer.

6.32.9 Electrical specification

6.32.9.1 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIS,SCL}$	Bit rates for TWIS[39]	100		400	kbps
$t_{TWIS,START}$	Time from PREPARERX/PREPARETX task to ready to receive/ transmit		1.5		µs
t_{TWIS,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIS,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIS,HD\_STA,100kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 100 kbps	5200			ns
$t_{TWIS,HD\_STA,400kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 400 kbps	1300			ns
$t_{TWIS,SU\_STO,100kbps}$	TWI slave setup time from SCL high to STOP condition, 100 kbps	5200			ns
$t_{TWIS,SU\_STO,400kbps}$	TWI slave setup time from SCL high to STOP condition, 400 kbps	1300			ns
$t_{TWIS,BUF,100kbps}$	TWI slave bus free time between STOP and START conditions, 100 kbps		4700		ns
$t_{TWIS,BUF,400kbps}$	TWI slave bus free time between STOP and START conditions, 400 kbps		1300		ns

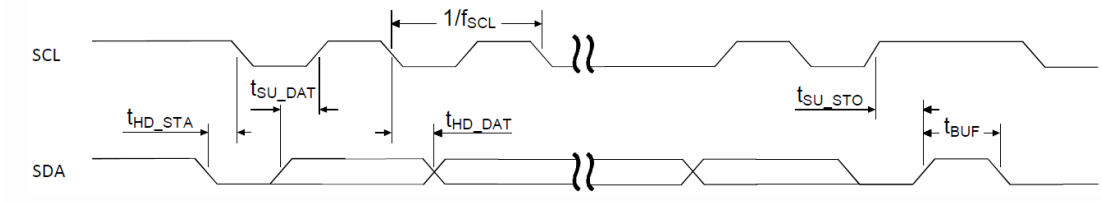

Figure 186: TWIS timing diagram, 1 byte transaction

6.33 UART — Universal asynchronous receiver/ transmitter

[39] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

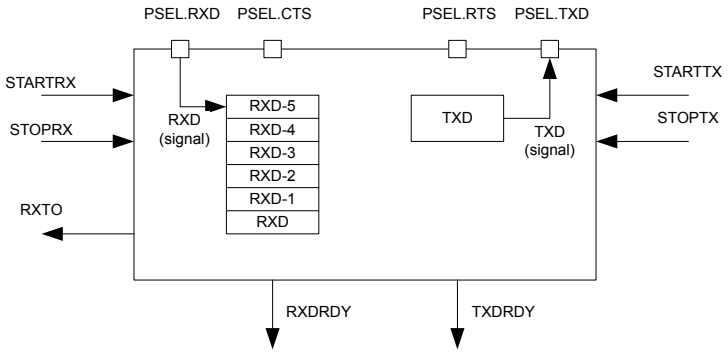

Figure 187: UART configuration

6.33.1 Functional description

Listed here are the main features of UART.

The UART implements support for the following features:

- Full-duplex operation
- Automatic flow control
- Parity checking and generation for the 9th data bit

As illustrated in UART configuration on page 455, the UART uses the TXD and RXD registers directly to transmit and receive data. The UART uses one stop bit.

Note: External crystal oscillator must be enabled to obtain sufficient clock accuracy for stable communication. See CLOCK — Clock control on page 80 for more information.

6.33.2 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UART are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

If the CONNECT field of a PSEL.xxx register is set to Disconnected, the associated UART signal will not be connected to any physical pin. The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UART is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.CTS, PSEL.RTS and PSEL.TXD must only be configured when the UART is disabled.

To secure correct signal levels on the pins by the UART when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in Pin configuration on page 455.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

UART pin	Direction	Output value
RXD	Input	Not applicable
CTS	Input	Not applicable
RTS	Output	1
TXD	Output	1

Table 124: GPIO configuration

6.33.3 Shared resources

The UART shares registers and other resources with other peripherals that have the same ID as the UART.

Therefore, you must disable all peripherals that have the same ID as the UART before the UART can be configured and used. Disabling a peripheral that has the same ID as the UART will not reset any of the registers that are shared with the UART. It is therefore important to configure all relevant UART registers explicitly to ensure that it operates correctly.

See the Instantiation table in Instantiation on page 22 for details on peripherals and their IDs.

6.33.4 Transmission

A UART transmission sequence is started by triggering the STARTTX task.

Bytes are transmitted by writing to the TXD register. When a byte has been successfully transmitted the UART will generate a TXDRDY event after which a new byte can be written to the TXD register. A UART transmission sequence is stopped immediately by triggering the STOPTX task.

If flow control is enabled a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in UART transmission on page 456. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended. For more information, see Suspending the UART on page 457.

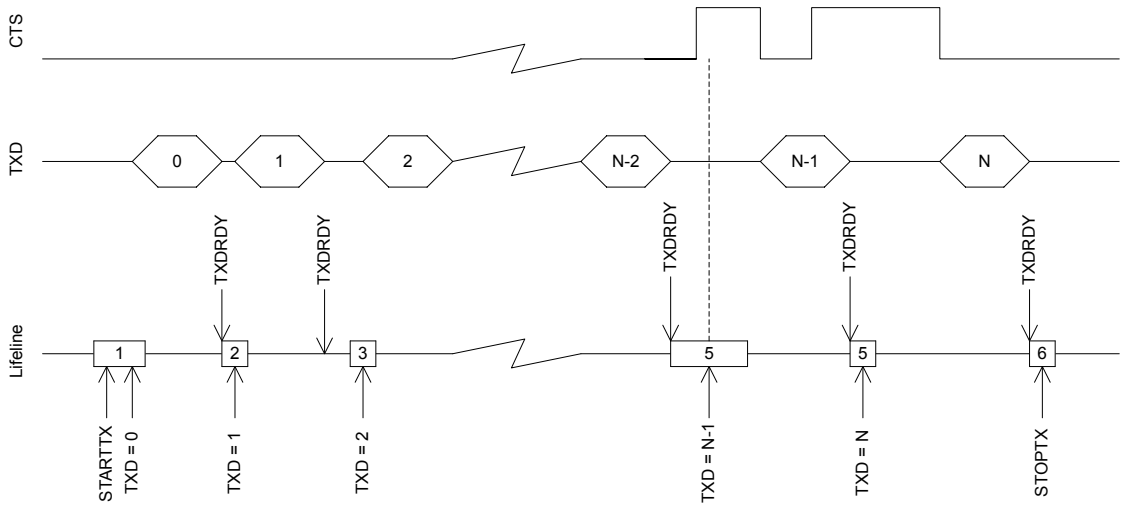

Figure 188: UART transmission

6.33.5 Reception

A UART reception sequence is started by triggering the STARTRX task.

The UART receiver chain implements a FIFO capable of storing six incoming RXD bytes before data is overwritten. Bytes are extracted from this FIFO by reading the RXD register. When a byte is extracted from the FIFO a new byte pending in the FIFO will be moved to the RXD register. The UART will generate an RXDRDY event every time a new byte is moved to the RXD register.

When flow control is enabled, the UART will deactivate the RTS signal when there is only space for four more bytes in the receiver FIFO. The counterpart transmitter is therefore able to send up to four bytes after the RTS signal is deactivated before data is being overwritten. To prevent overwriting data in the FIFO, the counterpart UART transmitter must therefore make sure to stop transmitting data within four bytes after the RTS line is deactivated.

The RTS signal will first be activated again when the FIFO has been emptied, that is, when all bytes in the FIFO have been read by the CPU, see UART reception on page 457.

The RTS signal will also be deactivated when the receiver is stopped through the STOPRX task as illustrated in UART reception on page 457. The UART is able to receive four to five additional bytes if they are sent in succession immediately after the RTS signal has been deactivated. This is possible because the UART is, even after the STOPRX task is triggered, able to receive bytes for an extended period of time dependent on the configured baud rate. The UART will generate a receiver timeout event (RXTO) when this period has elapsed.

To prevent loss of incoming data the RXD register must only be read one time following every RXDRDY event.

To secure that the CPU can detect all incoming RXDRDY events through the RXDRDY event register, the RXDRDY event register must be cleared before the RXD register is read. The reason for this is that the UART is allowed to write a new byte to the RXD register, and therefore can also generate a new event, immediately after the RXD register is read (emptied) by the CPU.

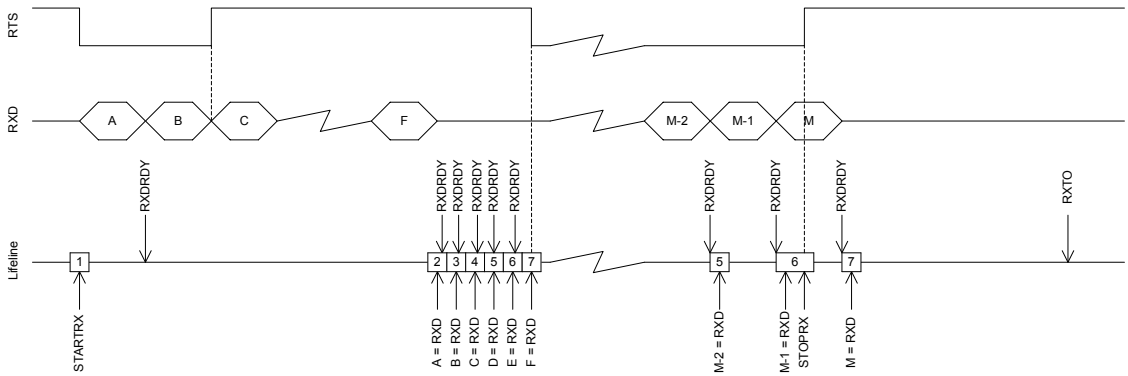

Figure 189: UART reception

As indicated in occurrence 2 in the figure, the RXDRDY event associated with byte B is generated first after byte A has been extracted from RXD.

6.33.6 Suspending the UART

The UART can be suspended by triggering the SUSPEND task.

SUSPEND will affect both the UART receiver and the UART transmitter, i.e. the transmitter will stop transmitting and the receiver will stop receiving. UART transmission and reception can be resumed, after being suspended, by triggering STARTTX and STARTRX respectively.

Following a SUSPEND task, an ongoing TXD byte transmission will be completed before the UART is suspended.

When the SUSPEND task is triggered, the UART receiver will behave in the same way as it does when the STOPRX task is triggered.

6.33.7 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

6.33.8 Using the UART without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

6.33.9 Parity configuration

When parity is enabled, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

6.33.10 Registers

Base address	Peripheral	Instance	Description	Configuration	
0x40002000	UART	UART0	Universal asynchronous receiver/ transmitter		Deprecated

Table 125: Instances

Register	Offset	Description
TASKS_STARTRX	0x000	Start UART receiver
TASKS_STOPRX	0x004	Stop UART receiver
TASKS_STARTTX	0x008	Start UART transmitter
TASKS_STOPTX	0x00C	Stop UART transmitter
TASKS_SUSPEND	0x01C	Suspend UART
EVENTS_CTS	0x100	CTS is activated (set low). Clear To Send.
EVENTS_NCTS	0x104	CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108	Data received in RXD
EVENTS_TXDRDY	0x11C	Data sent from TXD
EVENTS_ERROR	0x124	Error detected
EVENTS_RXTO	0x144	Receiver timeout
SHORTS	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x480	Error source
ENABLE	0x500	Enable UART
PSEL.RTS	0x508	Pin select for RTS
PSEL.TXD	0x50C	Pin select for TXD
PSEL.CTS	0x510	Pin select for CTS
PSEL.RXD	0x514	Pin select for RXD
RXD	0x518	RXD register
TXD	0x51C	TXD register
BAUDRATE	0x524	Baud rate. Accuracy depends on the HFCLK source selected.
CONFIG	0x56C	Configuration of parity and hardware flow control

Table 126: Register overview

6.33.10.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CTS_STARTRX			Shortcut between CTS event and STARTRX task See EVENTS_CTS and TASKS_STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	NCTS_STOPRX			Shortcut between NCTS event and STOPRX task See EVENTS_NCTS and TASKS_STOPRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.33.10.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to enable interrupt for CTS event See EVENTS_CTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to enable interrupt for NCTS event See EVENTS_NCTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to enable interrupt for RXDRDY event See EVENTS_RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXDRDY			Write '1' to enable interrupt for TXDRDY event See EVENTS_TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXTO			Write '1' to enable interrupt for RXTO event See EVENTS_RXTO
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled

6.33.10.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to disable interrupt for CTS event See EVENTS_CTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to disable interrupt for NCTS event See EVENTS_NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to disable interrupt for RXDRDY event See EVENTS_RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXDRDY			Write '1' to disable interrupt for TXDRDY event See EVENTS_TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXTO			Write '1' to disable interrupt for RXTO event See EVENTS_RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.33.10.4 ERRORSRC

Address offset: 0x480

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A start bit is received while the previous data still lies in RXD. (Previous data is lost.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
B	RW	PARITY			Parity error A character with bad parity is received, if HW parity check is enabled.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
C	RW	FRAMING			Framing error occurred A valid stop bit is not detected on the serial data input after all bits in a character have been received.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
D	RW	BREAK			Break condition The serial data input is '0' for longer than the length of a data frame. (The data frame length is 10 bits without parity bit, and 11 bits with parity bit.).
			NotPresent	0	Read: error not present
			Present	1	Read: error present

6.33.10.5 ENABLE

Address offset: 0x500

Enable UART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable UART
			Disabled	0	Disable UART
			Enabled	4	Enable UART

6.33.10.6 PSEL.RTS

Address offset: 0x508

Pin select for RTS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.33.10.7 PSEL.TXD

Address offset: 0x50C

Pin select for TXD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.33.10.8 PSEL.CTS

Address offset: 0x510

Pin select for CTS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.33.10.9 PSEL.RXD

Address offset: 0x514

Pin select for RXD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.33.10.10 RXD

Address offset: 0x518

RXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RXD			RX data received in previous transfers, double buffered

6.33.10.11 TXD

Address offset: 0x51C

TXD register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	TXD			TX data to be transferred

6.33.10.12 BAUDRATE

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	BAUDRATE			Baud rate
			Baud1200	0x0004F000	1200 baud (actual rate: 1205)
			Baud2400	0x0009D000	2400 baud (actual rate: 2396)
			Baud4800	0x0013B000	4800 baud (actual rate: 4808)
			Baud9600	0x00275000	9600 baud (actual rate: 9598)
			Baud14400	0x003B0000	14400 baud (actual rate: 14414)
			Baud19200	0x004EA000	19200 baud (actual rate: 19208)
			Baud28800	0x0075F000	28800 baud (actual rate: 28829)
			Baud31250	0x00800000	31250 baud
			Baud38400	0x009D5000	38400 baud (actual rate: 38462)
			Baud56000	0x00E50000	56000 baud (actual rate: 55944)

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x04000000				**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value				Description																											
			Baud57600	0x00EBF000				57600 baud (actual rate: 57762)																											
			Baud76800	0x013A9000				76800 baud (actual rate: 76923)																											
			Baud115200	0x01D7E000				115200 baud (actual rate: 115942)																											
			Baud230400	0x03AFB000				230400 baud (actual rate: 231884)																											
			Baud250000	0x04000000				250000 baud																											
			Baud460800	0x075F7000				460800 baud (actual rate: 470588)																											
			Baud921600	0x0EBED000				921600 baud (actual rate: 941176)																											
			Baud1M	0x10000000				1Mega baud																											

6.33.10.13 CONFIG

Address offset: 0x56C

Configuration of parity and hardware flow control

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																B	B	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	RW	Field	Value ID	Value				Description																											
A	RW	HWFC						Hardware flow control																											
			Disabled	0				Disabled																											
			Enabled	1				Enabled																											
B	RW	PARITY						Parity																											
			Excluded	0x0				Exclude parity bit																											
			Included	0x7				Include parity bit																											

6.33.11 Electrical specification

6.33.11.1 UART electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
f_{UART}	Baud rate for UART[40].			1000	kbps
$t_{UART,CTSH}$	CTS high time	1			µs
$t_{UART,START}$	Time from STARTRX/STARTTX task to transmission started		1		µs

6.34 UARTE — Universal asynchronous receiver/transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA (UARTE) offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in hardware at a rate up to 1 Mbps, and EasyDMA data transfer from/to RAM.

Listed here are the main features for UARTE:

- Full-duplex operation
- Automatic hardware flow control

[40] High baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

- Optional even parity bit checking and generation
- EasyDMA
- Up to 1 Mbps baudrate
- Return to IDLE between transactions supported (when using HW flow control)
- One or two stop bit
- Least significant bit (LSB) first

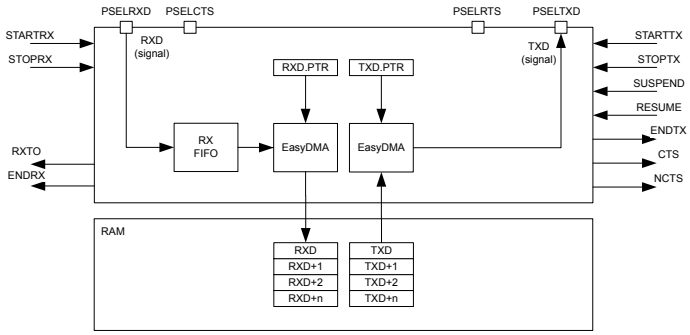

Figure 190: UARTE configuration

The GPIOs used for each UART interface can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

Note: External crystal oscillator must be enabled to obtain sufficient clock accuracy for stable communication. See CLOCK — Clock control on page 80 for more information.

6.34.1 EasyDMA

The UARTE implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 19 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The ENDRX/ENDTX event indicates that EasyDMA has finished accessing respectively the RX/TX buffer in RAM.

6.34.2 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to TXD.PTR, and the number of bytes in the RAM buffer to TXD.MAXCNT. The UARTE transmission is started by triggering the STARTTX task.

After each byte has been sent over the TXD line, a TXDRDY event will be generated.

When all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have been transmitted, the UARTE transmission will end automatically and an ENDTX event will be generated.

A UARTE transmission sequence is stopped by triggering the STOPTX task, a TXSTOPPED event will be generated when the UARTE transmitter has stopped.

If the ENDTX event has not already been generated when the UARTE transmitter has come to a stop, the UARTE will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

If flow control is enabled through the HWFC field in the CONFIG register, a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in UARTE transmission on page 466. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended.

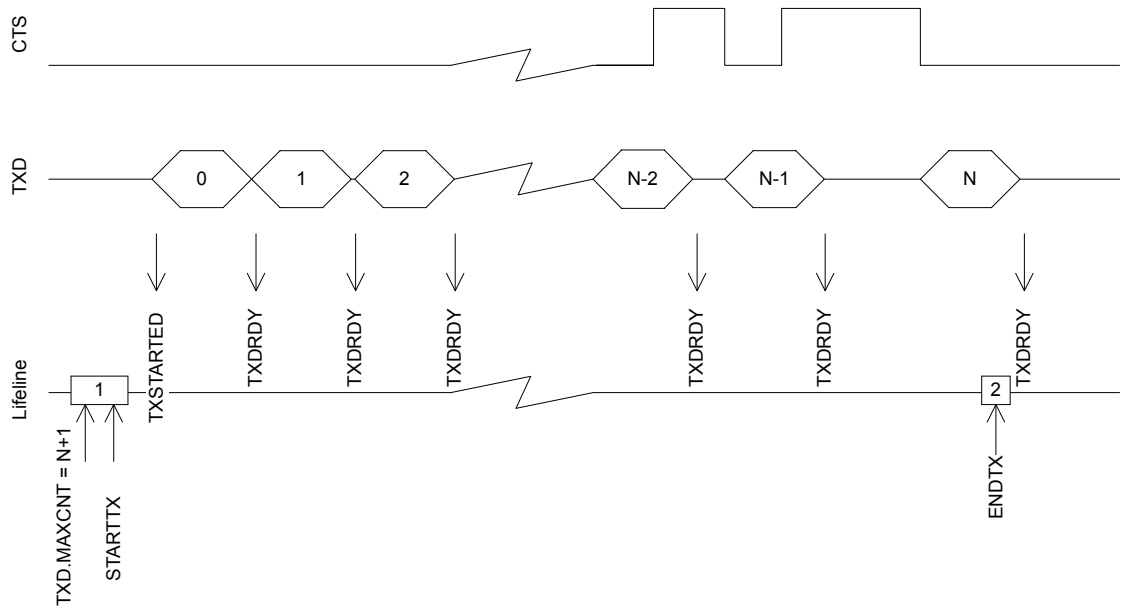

Figure 191: UARTE transmission

The UARTE transmitter will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTTX or after it has been stopped via STOPTX and the TXSTOPPED event has been generated. See POWER — Power supply on page 60 for more information about power modes.

6.34.3 Reception

The UARTE receiver is started by triggering the STARTRX task. The UARTE receiver is using EasyDMA to store incoming data in an RX buffer in RAM.

The RX buffer is located at the address specified in the RXD.PTR register. The RXD.PTR register is double-buffered and it can be updated and prepared for the next STARTRX task immediately after the RXSTARTED event is generated. The size of the RX buffer is specified in the RXD.MAXCNT register and the UARTE will generate an ENDRX event when it has filled up the RX buffer, see UARTE reception on page 467.

For each byte received over the RXD line, an RXDRDY event will be generated. This event is likely to occur before the corresponding data has been transferred to Data RAM.

The RXD.AMOUNT register can be queried following an ENDRX event to see how many new bytes have been transferred to the RX buffer in RAM since the previous ENDRX event.

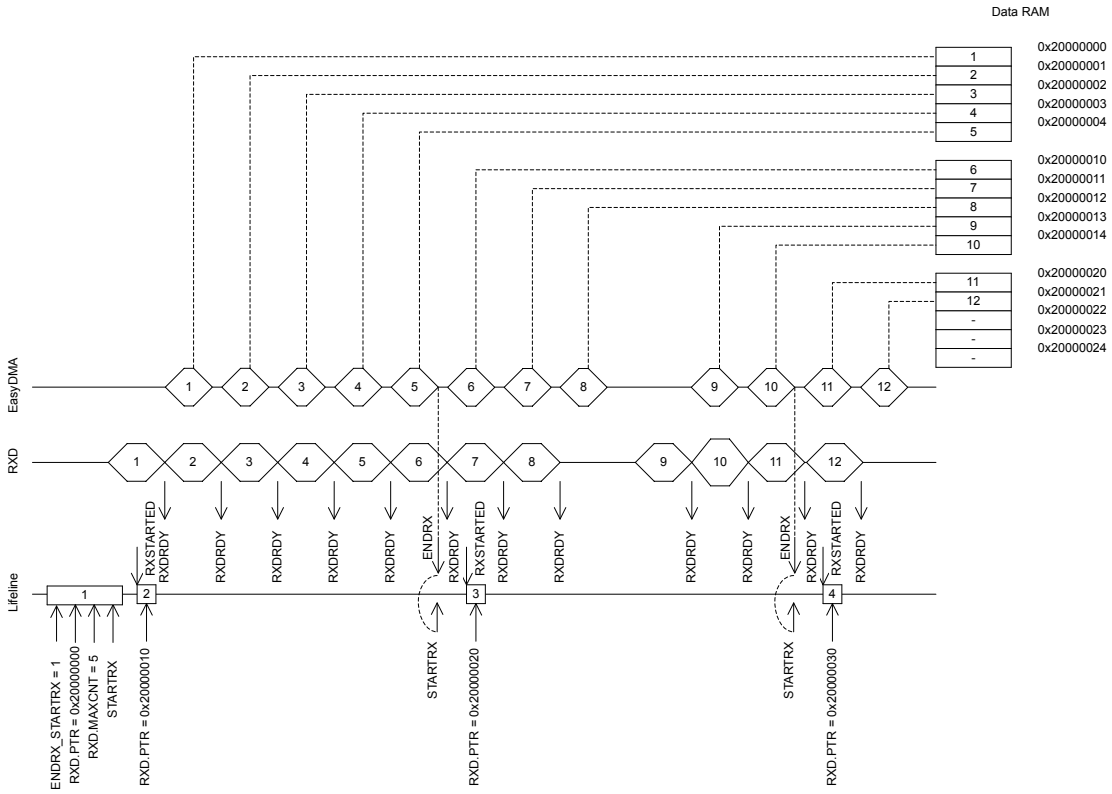

Figure 192: UARTE reception

The UARTE receiver is stopped by triggering the STOPRX task. An RXTO event is generated when the UARTE has stopped. The UARTE will make sure that an impending ENDRX event will be generated before the RXTO event is generated. This means that the UARTE will guarantee that no ENDRX event will be generated after RXTO, unless the UARTE is restarted or a FLUSHRX command is issued after the RXTO event is generated.

Important: If the ENDRX event has not already been generated when the UARTE receiver has come to a stop, which implies that all pending content in the RX FIFO has been moved to the RX buffer, the UARTE will generate the ENDRX event explicitly even though the RX buffer is not full. In this scenario the ENDRX event will be generated before the RXTO event is generated.

To be able to know how many bytes have actually been received into the RX buffer, the CPU can read the RXD.AMOUNT register following the ENDRX event or the RXTO event.

The UARTE is able to receive up to four bytes after the STOPRX task has been triggered as long as these are sent in succession immediately after the RTS signal is deactivated. This is possible because after the RTS is deactivated the UARTE is able to receive bytes for an extended period equal to the time it takes to send 4 bytes on the configured baud rate.

After the RXTO event is generated the internal RX FIFO may still contain data, and to move this data to RAM the FLUSHRX task must be triggered. To make sure that this data does not overwrite data in the RX buffer, the RX buffer should be emptied or the RXD.PTR should be updated before the FLUSHRX task is triggered. To make sure that all data in the RX FIFO is moved to the RX buffer, the RXD.MAXCNT register must be set to RXD.MAXCNT > 4, see UARTE reception with forced stop via STOPRX on page 468. The UARTE will generate the ENDRX event after completing the FLUSHRX task even if the RX FIFO was empty or if the RX buffer does not get filled up. To be able to know how many bytes have actually been received into the RX buffer in this case, the CPU can read the RXD.AMOUNT register following the ENDRX event.

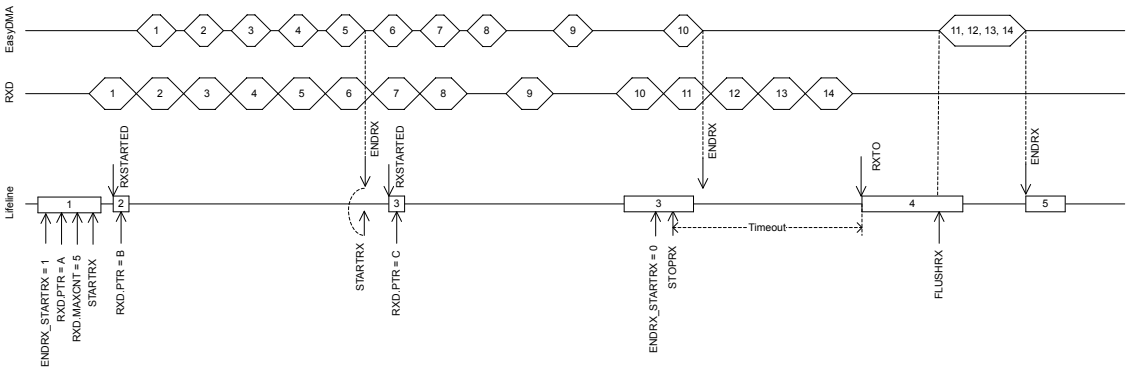

Figure 193: UARTE reception with forced stop via STOPRX

If HW flow control is enabled through the HWFC field in the CONFIG register, the RTS signal will be deactivated when the receiver is stopped via the STOPRX task or when the UARTE is only able to receive four more bytes in its internal RX FIFO.

With flow control disabled, the UARTE will function in the same way as when the flow control is enabled except that the RTS line will not be used. This means that no signal will be generated when the UARTE has reached the point where it is only able to receive four more bytes in its internal RX FIFO. Data received when the internal RX FIFO is filled up, will be lost.

The UARTE receiver will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTRX or after it has been stopped via STOPRX and the RXTO event has been generated. See POWER — Power supply on page 60 for more information about power modes.

6.34.4 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

An ERROR event will not stop reception. If the error was a parity error, the received byte will still be transferred into Data RAM, and so will following incoming bytes. If there was a framing error (wrong stop bit), that specific byte will NOT be stored into Data RAM, but following incoming bytes will.

6.34.5 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

6.34.6 Parity and stop bit configuration

When parity is enabled through the PARITY field in the CONFIG register, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

The amount of stop bits can be configured through the STOP field in the CONFIG register.

6.34.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOPTX and STOPRX tasks may not be always needed (the peripheral might already be stopped), but if STOPTX and/or STOPRX is sent, software shall wait until the TXSTOPPED and/or RXTO event is received in response, before disabling the peripheral through the ENABLE register.

6.34.8 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UARTE are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UARTE is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.RTS, PSEL.RTS and PSEL.TXD must only be configured when the UARTE is disabled.

To secure correct signal levels on the pins by the UARTE when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 469.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

UARTE signal	UARTE pin	Direction	Output value
RXD	As specified in PSEL.RXD	Input	Not applicable
CTS	As specified in PSEL.CTS	Input	Not applicable
RTS	As specified in PSEL.RTS	Output	1
TXD	As specified in PSEL.TXD	Output	1

Table 127: GPIO configuration before enabling peripheral

6.34.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40002000	UARTE	UARTE0	Universal asynchronous receiver/ transmitter with EasyDMA, unit 0	
0x40028000	UARTE	UARTE1	Universal asynchronous receiver/ transmitter with EasyDMA, unit 1	

Table 128: Instances

Register	Offset	Description
TASKS_STARTRX	0x000	Start UART receiver
TASKS_STOPRX	0x004	Stop UART receiver
TASKS_STARTTX	0x008	Start UART transmitter
TASKS_STOPTX	0x00C	Stop UART transmitter
TASKS_FLUSHRX	0x02C	Flush RX FIFO into RX buffer
EVENTS_CTS	0x100	CTS is activated (set low). Clear To Send.
EVENTS_NCTS	0x104	CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108	Data received in RXD (but potentially not yet transferred to Data RAM)
EVENTS_ENDRX	0x110	Receive buffer is filled up
EVENTS_TXDRDY	0x11C	Data sent from TXD
EVENTS_ENDTX	0x120	Last TX byte transmitted
EVENTS_ERROR	0x124	Error detected
EVENTS_RXTO	0x144	Receiver timeout
EVENTS_RXSTARTED	0x14C	UART receiver has started
EVENTS_TXSTARTED	0x150	UART transmitter has started
EVENTS_TXSTOPPED	0x158	Transmitter stopped
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt

Register	Offset	Description
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x480	Error source Note : this register is read / write one to clear.
ENABLE	0x500	Enable UART
PSEL.RTS	0x508	Pin select for RTS signal
PSEL.TXD	0x50C	Pin select for TXD signal
PSEL.CTS	0x510	Pin select for CTS signal
PSEL.RXD	0x514	Pin select for RXD signal
BAUDRATE	0x524	Baud rate. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534	Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last transaction
TXD.PTR	0x544	Data pointer
TXD.MAXCNT	0x548	Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C	Number of bytes transferred in the last transaction
CONFIG	0x56C	Configuration of parity and hardware flow control

Table 129: Register overview

6.34.9.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	ENDRX_STARTRX			Shortcut between ENDRX event and STARTRX task See EVENTS_ENDRX and TASKS_STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDRX_STOPRX			Shortcut between ENDRX event and STOPRX task See EVENTS_ENDRX and TASKS_STOPRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.34.9.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CTS			Enable or disable interrupt for CTS event See EVENTS_CTS
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
B	RW	NCTS			Enable or disable interrupt for NCTS event See EVENTS_NCTS
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	RXDRDY			Enable or disable interrupt for RXDRDY event See EVENTS_RXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ENDRX			Enable or disable interrupt for ENDRX event See EVENTS_ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXDRDY			Enable or disable interrupt for TXDRDY event See EVENTS_TXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	ENDTX			Enable or disable interrupt for ENDTX event See EVENTS_ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	ERROR			Enable or disable interrupt for ERROR event See EVENTS_ERROR
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	RXTO			Enable or disable interrupt for RXTO event See EVENTS_RXTO
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	RXSTARTED			Enable or disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	TXSTARTED			Enable or disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	TXSTOPPED			Enable or disable interrupt for TXSTOPPED event See EVENTS_TXSTOPPED
			Disabled	0	Disable
			Enabled	1	Enable

6.34.9.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	L J I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to enable interrupt for CTS event See EVENTS_CTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to enable interrupt for NCTS event See EVENTS_NCTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to enable interrupt for RXDRDY event See EVENTS_RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to enable interrupt for ENDRX event See EVENTS_ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to enable interrupt for TXDRDY event See EVENTS_TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to enable interrupt for ENDTX event See EVENTS_ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to enable interrupt for ERROR event See EVENTS_ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to enable interrupt for RXTO event See EVENTS_RXTO
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to enable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to enable interrupt for TXSTARTED event See EVENTS_TXSTARTED

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					L J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to enable interrupt for TXSTOPPED event See EVENTS_TXSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.34.9.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					L J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to disable interrupt for CTS event See EVENTS_CTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to disable interrupt for NCTS event See EVENTS_NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to disable interrupt for RXDRDY event See EVENTS_RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to disable interrupt for ENDRX event See EVENTS_ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to disable interrupt for TXDRDY event See EVENTS_TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to disable interrupt for ENDTX event See EVENTS_ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	L J I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to disable interrupt for ERROR event See EVENTS_ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to disable interrupt for RXTO event See EVENTS_RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to disable interrupt for RXSTARTED event See EVENTS_RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to disable interrupt for TXSTARTED event See EVENTS_TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to disable interrupt for TXSTOPPED event See EVENTS_TXSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.34.9.5 ERRORSRC

Address offset: 0x480

Error source

Note : this register is read / write one to clear.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A start bit is received while the previous data still lies in RXD. (Previous data is lost.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
B	RW	PARITY			Parity error A character with bad parity is received, if HW parity check is enabled.
			NotPresent	0	Read: error not present
			Present	1	Read: error present

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
C	RW	FRAMING			Framing error occurred A valid stop bit is not detected on the serial data input after all bits in a character have been received.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
D	RW	BREAK			Break condition The serial data input is '0' for longer than the length of a data frame. (The data frame length is 10 bits without parity bit, and 11 bits with parity bit.).
			NotPresent	0	Read: error not present
			Present	1	Read: error present

6.34.9.6 ENABLE

Address offset: 0x500

Enable UART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable UARTE
			Disabled	0	Disable UARTE
			Enabled	8	Enable UARTE

6.34.9.7 PSEL.RTS

Address offset: 0x508

Pin select for RTS signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.34.9.8 PSEL.TXD

Address offset: 0x50C

Pin select for TXD signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.34.9.9 PSEL.CTS

Address offset: 0x510

Pin select for CTS signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.34.9.10 PSEL.RXD

Address offset: 0x514

Pin select for RXD signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.34.9.11 BAUDRATE

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	BAUDRATE			Baud rate
			Baud1200	0x0004F000	1200 baud (actual rate: 1205)
			Baud2400	0x0009D000	2400 baud (actual rate: 2396)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Baud4800	0x0013B000	4800 baud (actual rate: 4808)
			Baud9600	0x00275000	9600 baud (actual rate: 9598)
			Baud14400	0x003AF000	14400 baud (actual rate: 14401)
			Baud19200	0x004EA000	19200 baud (actual rate: 19208)
			Baud28800	0x0075C000	28800 baud (actual rate: 28777)
			Baud31250	0x00800000	31250 baud
			Baud38400	0x009D0000	38400 baud (actual rate: 38369)
			Baud56000	0x00E50000	56000 baud (actual rate: 55944)
			Baud57600	0x00EB0000	57600 baud (actual rate: 57554)
			Baud76800	0x013A9000	76800 baud (actual rate: 76923)
			Baud115200	0x01D60000	115200 baud (actual rate: 115108)
			Baud230400	0x03B00000	230400 baud (actual rate: 231884)
			Baud250000	0x04000000	250000 baud
			Baud460800	0x07400000	460800 baud (actual rate: 457143)
			Baud921600	0x0F000000	921600 baud (actual rate: 941176)
			Baud1M	0x10000000	1Mega baud

6.34.9.12 RXD.PTR

Address offset: 0x534

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.34.9.13 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

6.34.9.14 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

6.34.9.15 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.34.9.16 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

6.34.9.17 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

6.34.9.18 CONFIG

Address offset: 0x56C

Configuration of parity and hardware flow control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B B A
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ID	RW	Field	Value ID	Value	Description
A	RW	HWFC			Hardware flow control
			Disabled	0	Disabled
			Enabled	1	Enabled
B	RW	PARITY			Parity
			Excluded	0x0	Exclude parity bit
			Included	0x7	Include even parity bit
C	RW	STOP			Stop bits
			One	0	One stop bit
			Two	1	Two stop bits

6.34.10 Electrical specification

6.34.10.1 UARTE electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
f_{UARTE}	Baud rate for UARTE[41].			1000	kbps
$t_{UARTE,CTSH}$	CTS high time	1			µs
$t_{UARTE,START}$	Time from STARTRX/STARTTX task to transmission started		1		µs

6.35 USBD — Universal serial bus device

The USB device (USBD) controller implements a full speed USB device function that meets 2.0 revision of the USB specification.

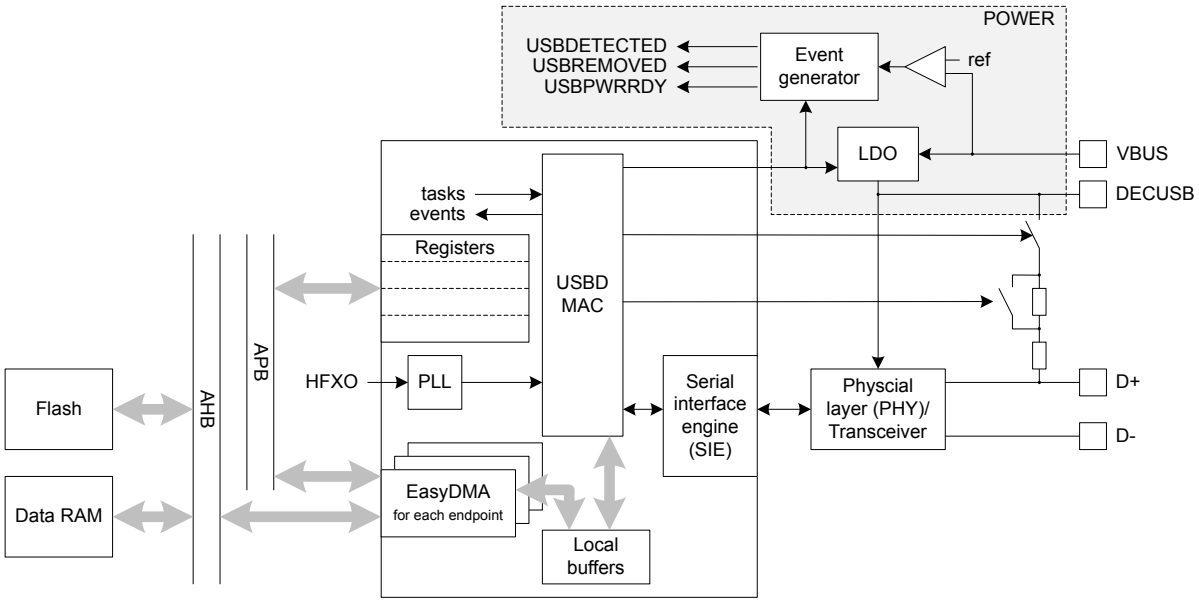

Figure 194: USB device block diagram

Listed here are the main features for USBD:

- Implements full-speed (12 Mbps) device fully compliant to Universal Serial Bus Specification Revision 2.0, including following engineering change notices (ECNs) issued by USB Implementers Forum:

[41] High baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

  - *Pull-up/pull-down Resistors ECN*
  - *5V Short Circuit Withstand Requirement Change ECN*
- USB device stack available in the Nordic SDK
- Integrated (on-chip) USB transceiver (PHY)
- Software controlled on-chip pull-up on D+
- Endpoints:
  - 2 control (1 IN, 1 OUT)
  - 14 bulk/interrupt (7 IN, 7 OUT)
  - 2 isochronous (1 IN, 1 OUT)
- Supports double buffering for isochronous (ISO) endpoints (IN/OUT)
- Supports USB suspend, resume, and remote wake-up
- 64 bytes buffer size for each bulk/interrupt endpoint
- Up to 1023 bytes buffer size for ISO endpoints
- EasyDMA for all data transfers

6.35.1 USB device states

The behavior of a USB device can be modelled through a state diagram.

The USB specification revision 2.0 (see *Chapter 9 USB Device Framework*) defines a number of states for a USB device, as illustrated below.

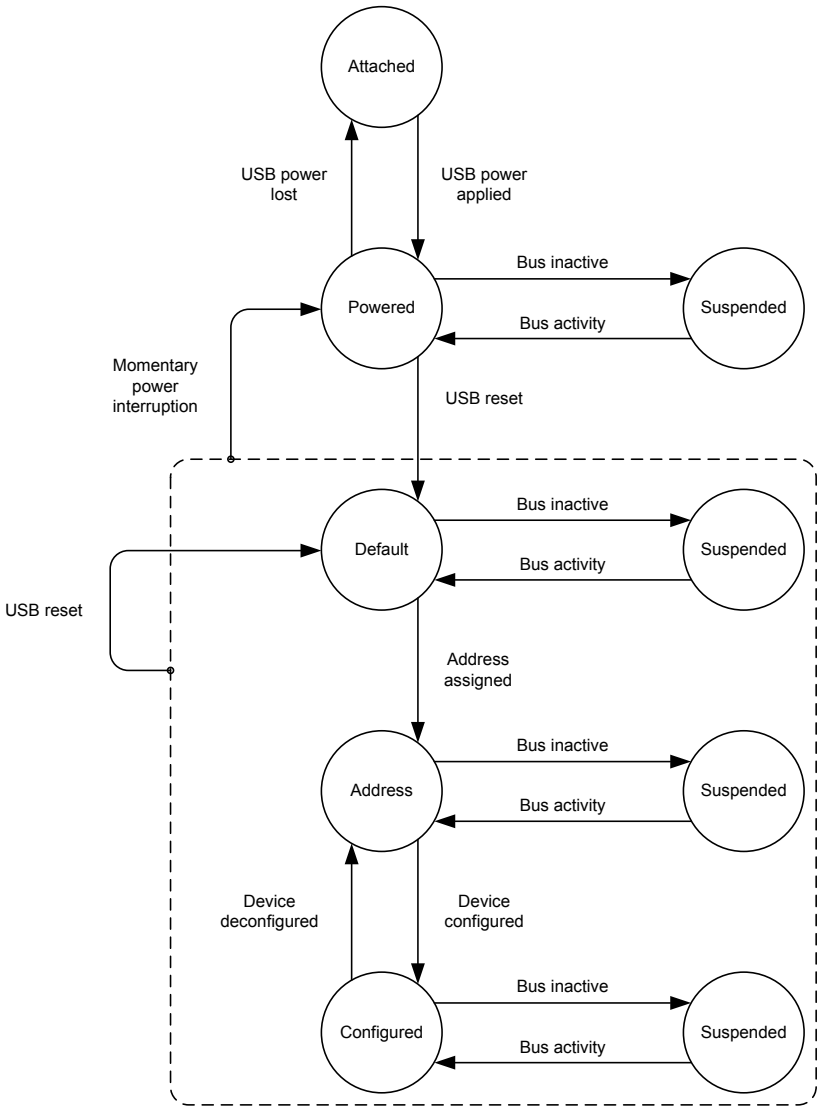

Figure 195: Device state diagram

The device must change state according to host-initiated traffic and USB bus states. It is up to the software to implement a state machine that matches the above definition. To detect the presence or absence of USB supply (VBUS), the POWER chapter defines two events, USBDETECTED and USBREMOVED, which can be used to implement the state machine.

As a general rule when implementing the software, the host behavior shall never be assumed to be predictable. In particular the sequence of commands received during an enumeration. The software shall always react to the current bus conditions or commands sent by the host.

6.35.2 USB terminology

The USB specification defines bus states, rather than logic levels on the D+ and D- lines.

For a full speed device, the bus state where the D+ line is high and the D- line is low is defined as the J state. The bus state where D+ is low and D- high is called the K state.

An idle bus, where D+ and D- lines are only polarized through the pull-up on D+ and pull-downs on the host side, will be in J state.

Both lines low are called SE0 (single-ended 0), and both lines high SE1 (single-ended 1).

6.35.3 USB pins

The USBD peripheral features a number of dedicated pins.

The dedicated USB pins can be grouped in two categories, signal and power. The signal pins consist of the D+ and D- pins, which are to be connected to the USB host. They are dedicated pins, and not available as standard GPIOs. The USBD implements the *5V Short Circuit Withstand ECN* meaning that these two pins are not 5 V tolerant.

The signal pins and the pull-up will operate only while VBUS is in its valid voltage range, and USBD is enabled through the ENABLE register. For details on the USB power supply and VBUS detection, see POWER.

See Pin assignments on page 524 for more information about the pinout.

6.35.4 USBD power-up sequence

The physical layer interface (PHY)/USB transceiver is powered separately from the rest of the device (VBUS pin), which has some implications on the USBD power-up sequence.

The device is not able to properly signal its presence to the USB host and handle traffic from the host, unless the PHY's power supply is enabled and stable. Turning the PHY's power supply on/off is directly linked to register ENABLE. The device provides events that help synchronizing software to the various steps during the power-up sequence.

To make sure that all resources in USBD are available and the dedicated USB voltage regulator stabilized, the following is recommended:

- Enable USBD after VBUS has been detected only
- Turn the USB pull-up on after:
  - USBPWRRDY event has occurred
  - USBEVENT has occurred, with the READY condition flagged in EVENTCAUSE

The following sequence chart illustrates a typical handling of VBUS power-up:

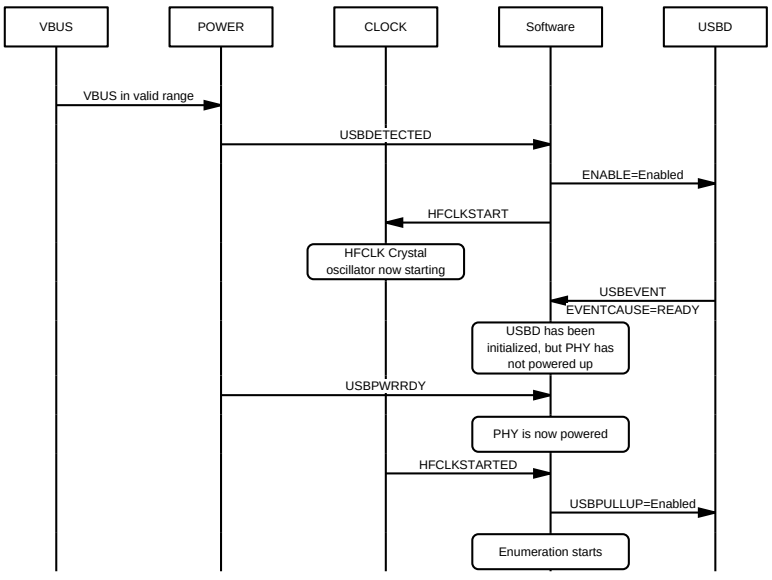

Figure 196: VBUS power-up sequence

Upon VBUS removal detection, signalled by the USBREMOVED event described in POWER, it is recommended to let on-going EasyDMA transfers finish (wait for the relevant ENDEPIN[n], ENDISOIN, ENDEPOUT[n] or ENDISOOUT event, see EasyDMA on page 485), before disabling USBD (by writing ENABLE=Disabled).

6.35.5 USB pull-up

The USB pull-up serves two purposes - it indicates to the host that the device is connected to the USB bus, and it indicates the device's speed capability.

When no pull-up is connected to the USB bus, the host sees both D+ and D- lines low, as they are pulled down on the host side by 15 kΩ resistors. The device is not seen by the host and hence in detached state, even though it could be physically connected to the host. USB specification does not allow to draw any current on VBUS in that situation.

When a full-speed device connects its 1.5 kΩ pull-up to D+, the host sees the corresponding line high. The device is then in the attached state. During the enumeration process, the host attempts to determine if the full-speed device also supports higher speeds and initiates communication with the device to further identify it. The USBD peripheral implemented in this device supports only full-speed (12 Mbps), and thus ignores the negotiation for higher speeds in accordance with the USB specification revision 2.0.

Register USBPULLUP provides means to connect or disconnect the pull-up on D+ under software control. This allows the software to control when USB enumeration takes place. It also allows to emulate a physical disconnect from the USB bus, for instance when re-enumeration is required. USBPULLUP has to be enabled to allow the USBD to handle USB traffic and generate appropriate events. This forbids the use of an external pull-up.

Note that disconnecting the pull-up through register USBPULLUP while connected to a host, will result in both D+ and D- lines to be pulled low by the host's pull-down resistors. However, as mentioned above, this will also inhibit the generation of the USBRESET event. The pull-up is disabled by default after a chip reset.

The pull-up shall only get connected after USBD has been enabled through register ENABLE. The USB pull-up value is automatically changed depending on the bus activity, as specified in *Resistor ECN* which amends the original USB specification version 2.0. The user does not have access to this function, it is handled in hardware.

While they should never be used in normal traffic activity, lines D+ and D- may at any time be forced into state specified in register DPDMVALUE by the task DPDMDRIVE. The DPDMNODRIVE task stops driving them, and PHY returns to normal operation.

6.35.6 USB reset

The USB specification defines a USB reset, which is not be confused with a chip reset. The USB reset is a normal USB bus condition, and is used as part of the enumeration sequence, it does not reset the chip.

The USB reset results from a single-ended low state (SE0) on lines D+/D- for a $t_{USB,DETRST}$ amount of time. Only the host is allowed to drive a USB reset condition on the bus. The UBSD peripheral automatically interprets a SE0 longer than $t_{USB,DETRST}$ as a USB reset. When the device detects a USB reset and generates a USBRESET event, the device USB stack and related parts of the application shall re-initialize themselves, and go back to the default state.

Some of the registers in the USBD peripheral get automatically reset to a known state, in particular all data endpoints are disabled and the USBADDR reset to 0.

After the device has connected to the USB bus (i.e. after VBUS is applied), the device shall not respond to any traffic from the time the pull-up is enabled until it has seen a USB reset condition. This is automatically ensured by the USBD.

After a USB reset, the device shall be fully responsive after at most T_{RSTRCY} (according to chapter 7 in the USB specification). Software shall take into account this time that takes the hardware to recover from a USB reset condition.

6.35.7 USB suspend and resume

Normally, the host will maintain activity on the USB at least every millisecond according to USB specification. A USB device will enter suspend when there is no activity on the bus (idle) for a given time. The device will resume operation when it receives any non idle signalling.

To signal that the device shall go into low power mode (suspend), the host stops activity on the USB bus, which becomes idle. Only the device pull-up and host pull-downs act on D+ and D-, and the bus is thus kept at a constant J state. It is up to the device to detect this lack of activity, and enter the low power mode (suspend) within a specified time.

The USB host can decide to suspend or resume USB activity at any time. If remote wake-up is enabled, the device may signal to the host to resume from suspend.

6.35.7.1 Entering suspend

The USBD peripheral automatically detects lack of activity for more than a defined amount of time, and performs steps needed to enter suspend.

When no activity has been detected for longer than $t_{USB,SUSPEND}$, the USBD generates the USBEVENT event with SUSPEND bit set in register EVENTCAUSE. The software shall ensure that the current drawn from the USB supply line VBUS is within the specified limits before T_{2SUSP}, as defined in chapter 7 of the USB specification. In order to reduce idle current of USBD, the software must explicitly place the USBD in low power mode through writing `LowPower` to register LOWPOWER.

In order to save power, and provided that no other peripheral needs it, the crystal oscillator (HFXO) in CLOCK may be disabled by software during the USB suspend, while the USB pull-up is disconnected, or when VBUS is not present. Software must explicitly enable it at any other time. The USBD will not be able to respond to USB traffic unless HFXO is enabled and stable.

6.35.7.2 Host-initiated resume

Once the host resumes the bus activity, it has to be responsive to incoming requests on the USB bus within the time T_{RSMRCY} (as defined in chapter 7 of the USB specification) and revert to normal power consumption mode.

If the host resumes bus activity with or without a RESUME condition (in other words: bus activity is defined as any non-J state), the USBD peripheral will generate a USBEVENT event, with RESUME bit set in register EVENTCAUSE. If the host resumes bus activity simply by restarting sending frames, the USBD peripheral will generate SOF events.

6.35.7.3 Device-initiated remote wake-up

Assuming the remote wake-up is supported by the device and enabled by the host, the device can request the host to resume from suspend if wake-up condition is met.

To do so, the HFXO needs to be enabled first. After waking up the HFXO, the software must bring USBD out of the low power mode and into the normal power consumption mode through writing `ForceNormal` in register LOWPOWER. It can then instruct the USBD peripheral to drive a RESUME condition (K state) on the USB bus by triggering the DPDMDRIVE task, and hence attempt to wake up the host. By choosing `Resume` in DPDMVALUE, the duration of the RESUME state is under hardware control ($t_{USB,DRIVEK}$). By choosing `J` or `K`, the duration of that state is under software control (the J or K state is maintained until a DPDMNODRIVE task is triggered) and has to meet T_{DRSMUP} as specified in USB specification chapter 7.

Upon writing the `ForceNormal` in register LOWPOWER, a USBEVENT event is generated with the USBWUALLOWED bit set in register EVENTCAUSE.

The value in register DPDMVALUE on page 513 will only be captured and used when the DPDMDRIVE task is triggered. This value defines the state the bus will be forced into after the DPDMDRIVE task.

Note that the device shall ensure that it does not initiate a remote wake-up request before T_{WTRSM} (according to USB specification chapter 7) after the bus has entered idle state. Using the recommended

resume value in DPDMVALUE (rather than K) takes care of this, and postpones the RESUME state accordingly.

6.35.8 EasyDMA

The USBD peripheral includes EasyDMA, so USB buffers are located in Data RAM .

Each endpoint has an associated set of registers, tasks and events. EasyDMA and traffic on USB are tightly related. A number of events provide insight of what is happening on the USB bus, and a number of tasks allow to somewhat automate response to the traffic.

> **Note:** Endpoint 0 (IN and OUT) are implemented as control endpoint. For more information, see Control transfers on page 486.

Registers

Enabling endpoints is controlled through the EPINEN and EPOUTEN registers.

The following registers define the address of the buffer in Data RAM for a specific IN or OUT endpoint:

- EPIN[n].PTR, (n=0..7)
- EPOUT[n].PTR, (n=0..7)
- ISOIN.PTR
- ISOOUT.PTR

The following registers define the amount of bytes to be sent on USB for next transaction:

- EPIN[n].MAXCNT, (n=0..7)
- ISOIN.MAXCNT

The following registers define the length of the buffer (in bytes) for next transfer of incoming data:

- EPOUT[n].MAXCNT, (n=1..7)
- ISOOUT.MAXCNT

Since the host decides how many bytes are sent over USB, the MAXCNT value can be copied from register SIZE.EPOUT[n] (n=1..7) or register SIZE.ISOOUT.

Register EPOUT[0].MAXCNT defines the length of the OUT buffer (in bytes) for the control endpoint 0. If the USB host does not misbehave, register SIZE.EPOUT[0] will indicate the same value as `MaxPacketSize` from the device descriptor or `wLength` from the SETUP command, whichever the smallest.

The .AMOUNT registers indicate how many bytes actually have been transferred over EasyDMA during the last transfer.

Stalling bulk/interrupt endpoints is controlled through the EPSTALL register.

> **Note:** Due to USB specification requirements, the effect of the stalling control endpoint 0 may be overridden by hardware, in particular when a new SETUP token is received.

EasyDMA will not copy the SETUP data to Data RAM (it will only transfer data from the data stage). Setup data is available as separate registers in the USBD peripheral:

- BMREQUESTTYPE
- BREQUEST
- WVALUEL
- WVALUEH
- WINDEXL
- WINDEXH

- WLENGTHL
- WLENGTHH

EVENTCAUSE register provides details on what caused a given USBEVENT event, for instance if a CRC error is detected during a transaction, or if bus activity stops or resumes.

Tasks

Tasks STARTEPIN[n], STARTEPOUT[n] (n=0..7), STARTISOIN and STARTISOOUT capture the values for .PTR and .MAXCNT registers. For IN endpoints, a transaction over USB gets automatically triggered when the EasyDMA transfer is complete. For OUT endpoints, it is up to software to allow the next transaction over USB. See the examples in Control transfers on page 486, Bulk and interrupt transactions on page 489 and Isochronous transactions on page 492.

For the control endpoint 0, OUT transactions are allowed through the EP0RCVOUT task. The EP0STATUS task allows a status stage to be initiated, and the EP0STALL task allows stalling further traffic (data or status stage) on the control endpoint.

Events

The STARTED event confirms that the values of the .PTR and .MAXCNT registers of the endpoints flagged in register EPSTATUS have been captured. Those can then be modified by software for the next transfer.

Events ENDEPIN[n], ENDEPOUT[n] (n=0..7), ENDISOIN and ENDISOOUT events indicate that the whole buffer in Data RAM has been consumed. The buffer can be accessed safely by the software.

Only a single EasyDMA transfer can take place in USBD at any time. Software must ensure that tasks STARTEPIN[n] (n=0..7), STARTISOIN, STARTEPOUT[n] (n=0..7) or STARTISOOUT are not triggered before events ENDEPIN[n] (n=0..7), ENDISOIN, ENDEPOUT[n] (n=0..7) or ENDISOOUT are received from an on-going transfer.

The EPDATA event indicates that a successful (acknowledged) data transaction has occurred on the data endpoint(s) flagged in register EPDATASTATUS. A successful (acknowledged) data transaction on endpoint 0 is signalled by the EP0DATADONE event.

At any time a USBEVENT event may be sent, with details provided in EVENTCAUSE register.

EP0SETUP event indicates that a SETUP token has been received on the control endpoint 0, and that the setup data is available in registers.

6.35.9 Control transfers

The USB specification mandates every USB device to implement endpoint 0 IN and OUT as control endpoints.

A control transfer consists of two or three stages:

- Setup stage
- Data stage (optional)
- Status stage

Each control transfer can be one of following types:

- Control read
- Control read no data
- Control write
- Control write no data

An EP0SETUP event indicates that the data in the setup stage (following the SETUP token) is available in registers.

The data in the data stage (following the IN or OUT token) is transferred from or to the desired location in Data RAM using EasyDMA.

Note: The control endpoint buffer size in Data RAM can be of any size in bytes, and there is no constraint to keep it 32-bit aligned.

After receiving the SETUP token, the USB controller will not accept (NAK) any incoming IN or OUT tokens until the software has finished decoding the command, determining the type of transfer, and preparing for the next stage (data or status) appropriately.

The software can choose to stall a command (in both data and status stages) through the EP0STALL task, for instance if the command is not supported, or its wValue, wIndex or wLength parameters are wrong. A stalled control read transfer is illustrated below, but the same mechanism (same tasks) applies to stalling a control write transfer (not illustrated):

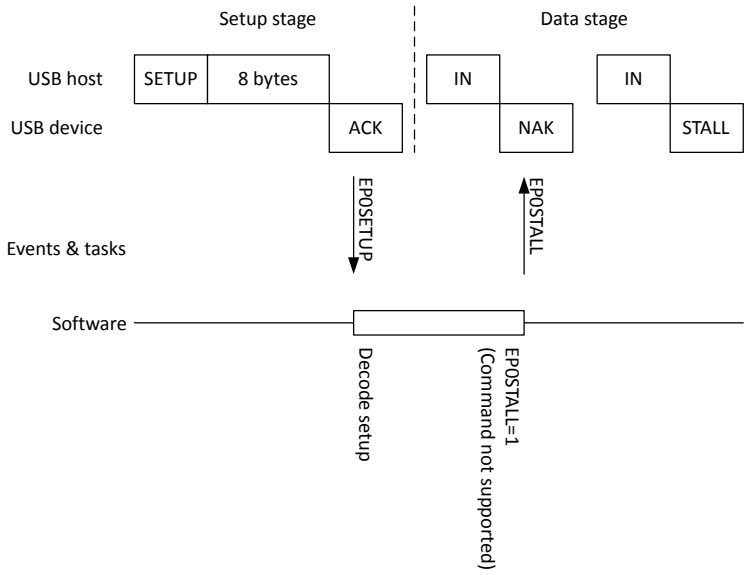

Figure 197: Control read gets stalled

See chapter 9 of the USB specification and relevant class specifications for rules on when to stall a command.

Note: The USBD peripheral handles the SetAddress transfer by itself. As a consequence, the software shall not process this command other than updating its state machine (see Device state diagram), nor initiate a status stage. If necessary, the address assigned by the host can be read out from the USBADDR register after the command has been processed.

6.35.9.1 Control read transfer

This section describes how the software behaves to respond to a control read transfer.

As mentioned earlier, the USB controller will not accept (NAK) any incoming IN tokens until software has finished decoding the command, determining the type of transfer, and preparing for the next stage (data or status) appropriately.

For a control read, transferring the data from Data RAM memory into USBD will trigger a valid, acknowledged (ACK) IN transaction on USB.

The software has to prepare EasyDMA by pointing to the buffer containing the data to be transferred. If no other EasyDMA transfers are on-going with USBD, the software can send the STARTEPIN0 task, which will initiate the data transfer and transaction on USB.

A STARTED event (with EPIN0 bit set in the EPSTATUS register) will be generated as soon as the EPIN[0].PTR and .MAXCNT registers have been captured. Software may then prepare them for the next data transaction.

An ENDEPIN[0] event will be generated when the data has been transferred from memory to the USBD peripheral.

Finally, an EP0DATADONE event will be generated when the data has been transmitted over USB and acknowledged by the host.

The software can then either prepare and transmit the next data transaction by repeating the above sequence, or initiate the status stage through the EP0STATUS task.

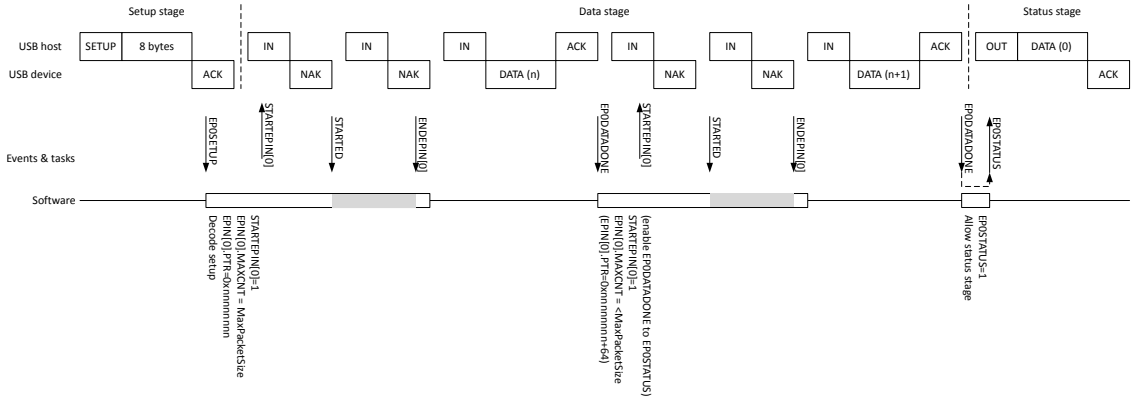

Figure 198: Control read transfer

Note the possibility to enable a shortcut from the EP0DATADONE event to the EP0STATUS task, typically if the data stage is expected to take a single transfer. If there is no data stage, the software can initiate the status stage through the EP0STATUS task right away, as illustrated below:

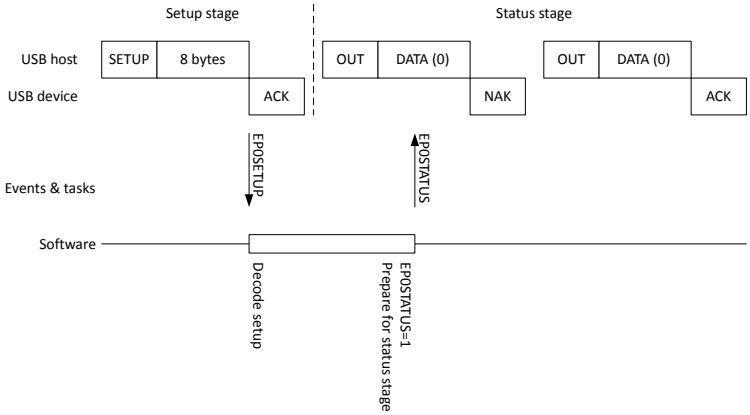

Figure 199: Control read no data transfer

6.35.9.2 Control write transfer

This section describes how the software responds to a control write transfer.

The software has to prepare EasyDMA by pointing to the buffer in Data RAM that shall contain the incoming data. If no other EasyDMA transfers are on-going with USBD, the software can then send the EP0RCVOUT task, which will make USBD acknowledge (ACK) the first OUT+DATA transaction from the host.

An EP0DATADONE event will be generated when a new OUT+DATA has been transmitted over USB, and is about to get acknowledged by the device.

A STARTED event (with EPOUT0 bit set in the EPSTATUS register) will be generated as soon as the EPOUT[0].PTR and .MAXCNT registers have been captured, after receiving the first transaction. Software may then prepare them for the next data transaction.

An ENDEPOUT[0] event will be generated when the data has been transferred from the USBD peripheral to Data RAM. The software can then either prepare to receive the next data transaction by repeating the above sequence, or initiate the status stage through the EP0STATUS task. Until then, further incoming OUT +DATA transactions get a NAK response by the device.

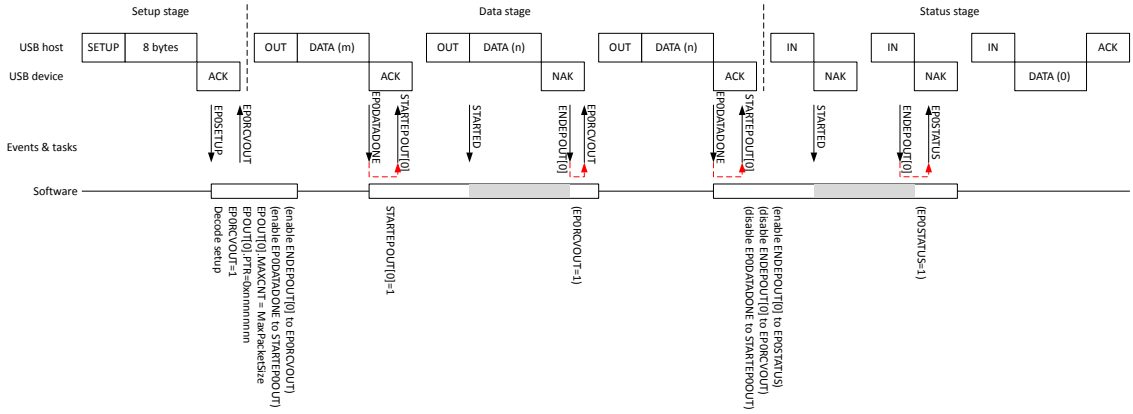

Figure 200: Control write transfer

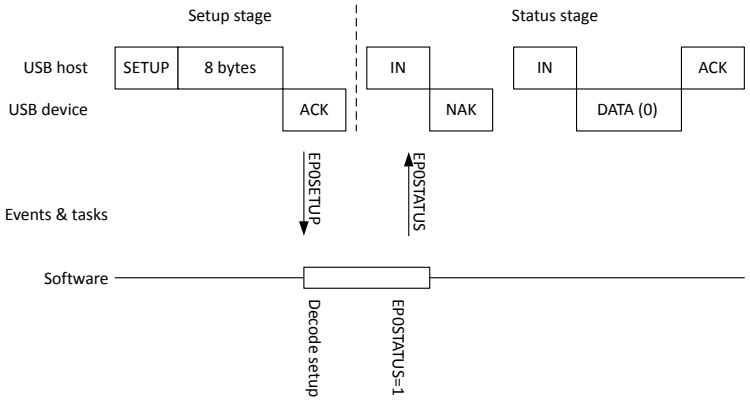

Figure 201: Control write no data transfer

6.35.10 Bulk and interrupt transactions

The USBD peripheral implements seven pairs of bulk/interrupt endpoints.

The bulk/interrupt endpoints have a fixed USB endpoint number, summarized in the table below.

Bulk endpoint #	USB IN endpoint	USB OUT endpoint
[1]	0x81	0x01
[2]	0x82	0x02
[3]	0x83	0x03
[4]	0x84	0x04
[5]	0x85	0x05
[6]	0x86	0x06
[7]	0x87	0x07

Table 130: Bulk/interrupt endpoint numbering

A bulk/interrupt transaction consists of a single data stage. Two consecutive, successful transactions are distinguished through alternating leading process ID (PID): DATA0 follows DATA1, DATA1 follows DATA0,

etc. A repeated transaction is detected by re-using the same PID as previous transaction, i.e DATA0 follows DATA0, or DATA1 follows DATA1.

The USBD controller automatically toggles DATA0/DATA1 PIDs for every bulk/interrupt transaction, and in general software does not need to care about it.

If an incoming data is corrupted (CRC does not match), the USBD controller automatically prevents DATA0/DATA1 from toggling, to request the host to resend the data.

In some specific cases, the software may want to force a data toggle (usually reset) on a specific IN endpoint, or force the expected toggle on an OUT endpoint, for instance as a consequence of the host issuing **`ClearFeature`**, **`SetInterface`** or selecting an alternate setting. Controlling the data toggle of data IN or OUT endpoint n (n=1..7) is done through register DTOGGLE.

The maximum size of a bulk/interrupt transaction in USB full-speed is 64 bytes, and it has to be a multiple of 4 bytes and 32-bit aligned in Data RAM. However, the amount of data bytes transmitted on the USB data endpoint can be of any size (up to 64 bytes).

When the transaction is done over USB, an EPDATA event is generated. The hardware will then automatically respond with NAK to all incoming IN tokens until the software is ready to send more data and has finished configuring the EasyDMA, started it, and the whole buffer content has been moved to USB controller (signalled by the ENDEPIN[n] event).

Each IN or OUT data endpoint has to be explicitly enabled by software through register EPINEN or EPOUTEN, according to the configuration declared by the device and selected by the host through the **`SetConfig`** command.

A disabled data endpoint will not respond to any traffic from the host. An enabled data endpoint will normally respond NAK or ACK (depending on the readiness of the buffers), or STALL (if configured in register EPSTALL), in which case the endpoint is asked to halt). The halted (or not) state of a given endpoint can be read back from register HALTED.EPIN[n] or HALTED.EPOUT[n]. The format of the returned 16-bit value can be copied as is as response to a **`GetStatusEndpoint`** request from the host.

Note that enabling or disabling an endpoint will not change its halted state. However, a USB reset will disable and clear the halted state of all data endpoints.

The control endpoint 0 IN and OUT can also be enabled and/or halted using the same mechanisms, but due to USB specification, receiving a SETUP will override its state.

6.35.10.1 Bulk and interrupt IN transaction

The host issues IN tokens to receive bulk/interrupt data. In order to send data, the software has to enable the endpoint and prepare an EasyDMA transfer on the desired endpoint.

Bulk/interrupt IN endpoints are enabled or disabled through their respective INn bit (n=1..7) in EPINEN register.

It is also possible to stall or un-stall an endpoint through the EPSTALL register.

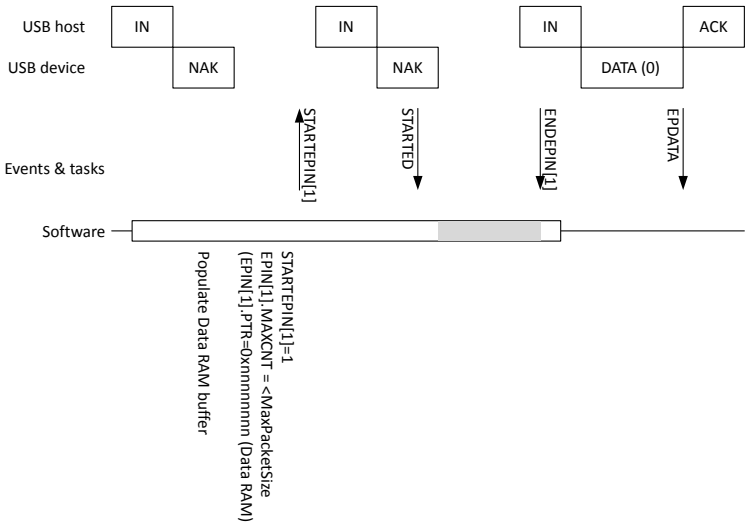

Figure 202: Bulk/interrupt IN transaction

It is possible (and in some situations it is required) to respond to an IN token with a zero-length data packet.

> **Note:** On many USB hosts, not responding (DATA+ACK or NAK) to three IN tokens on an interrupt endpoint would have the host disable that endpoint as a consequence. Re-enumerating the device (unplug-replug) may be required to restore functionality. Make sure that the relevant data endpoints are enabled for normal operation as soon as the device gets configured through a **`SetConfig`** request.

6.35.10.2 Bulk and interrupt OUT transaction

When the host wants to transmit bulk/interrupt data, it issues an OUT token (packet) followed by a DATA packet on a given endpoint n (n=1..7).

A NAK is returned until the software writes any value to register SIZE.EPOUT[n], indicating that the content of the local buffer can be overwritten. Upon receiving the next OUT+DATA transaction, an ACK is returned to the host while an EPDATA event is generated (and the EPDATASTATUS register flags are set to indicate on which endpoint this happened). Once the EasyDMA is prepared and enabled, by writing the EPOUT[n] registers and triggering the STARTEPOUT[n] task, the incoming data will be transferred to Data RAM. Until that transfer is finished, the hardware will automatically NAK any other incoming OUT+DATA packets. Only when the EasyDMA transfer is done (signalled by the ENDEPOUT[n] event), or as soon as any values are written by the software in register SIZE.EPOUT[n], the endpoint n will accept incoming OUT+DATA again.

It is allowed for the host to send zero-length data packets.

Bulk/interrupt OUT endpoints are enabled or disabled through their respective OUTn bit (n=1..7) in the EPOUTEN register. It is also possible to stall or un-stall an endpoint through the EPSTALL register.

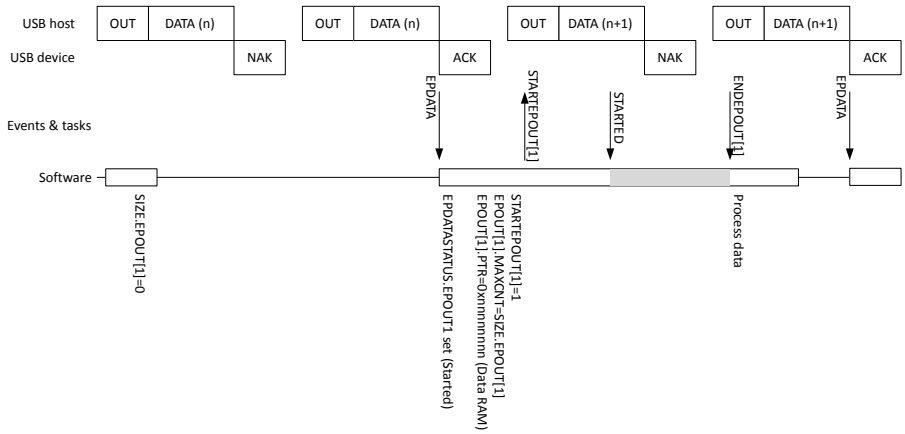

Figure 203: Bulk/interrupt OUT transaction

6.35.11 Isochronous transactions

The USBD peripheral implements isochronous (ISO) endpoints.

The ISO endpoints have a fixed USB endpoint number, summarized in the table below.

ISO endpoint #	USB IN endpoint	USB OUT endpoint
[0]	0x88	0x08

Table 131: Isochronous endpoint numbering

An isochronous transaction consists of a single, non-acknowledged data stage. The host sends out a start of frame at a regular interval (1 ms), and data follows IN or OUT tokens within each frame.

EasyDMA allows transferring ISO data directly from and to Data RAM. EasyDMA transfers must be initiated by the software, which can synchronize with the SOF (start of frame) events.

Because the timing of the start of frame is very accurate, the SOF event can be used for instance to synchronize a local timer through the SOF event and PPI. The SOF event gets synchronized to the 16 MHz clock prior to being made available to the PPI.

Every start of frame increments a free-running counter, which can be read by software through the FRAMECNTR register.

Each IN or OUT ISO data endpoint has to be explicitly enabled by software through register EPINEN or EPOUTEN, according to the configuration declared by the device and selected by the host through the **`SetConfig`** command. A disabled ISO IN data endpoint will not respond to any traffic from the host. A disabled ISO OUT data endpoint will ignore any incoming traffic from the host.

The USBD peripheral has an internal 1 kB buffer associated with ISO endpoints. The user can either allocate the full amount to the IN or the OUT endpoint, or split the buffer allocation between the two. This is done through register ISOSPLIT, which provides a number of pre-determined splits.

6.35.11.1 Isochronous IN transaction

When the host wants to receive isochronous (ISO) data, it issues an IN token on the isochronous endpoint.

After the data has been transferred using the EasyDMA, the USB controller on the isochronous IN endpoint responds to the IN token with the transferred data using the ISOIN.MAXCNT for the size of the packet.

The ISO IN data endpoint has to be explicitly enabled by software through the ISOIN0 bit in register EPINEN.

When an ISO IN endpoint is enabled and no data transferred with EasyDMA, the response of the USBD depends on the setting of the RESPONSE field in register ISOINCONFIG - it can either provide no response to an IN token or respond with a zero-length data.

Note: The maximum size of an ISO IN transfer in USB full-speed is 1023 bytes, and the data buffer in RAM has to be a multiple of 4 bytes 32-bit aligned in Data RAM. However, the amount of bytes transferred on the USB data endpoint can be of any size (up to 1023 bytes, if not shared with an OUT ISO endpoint).

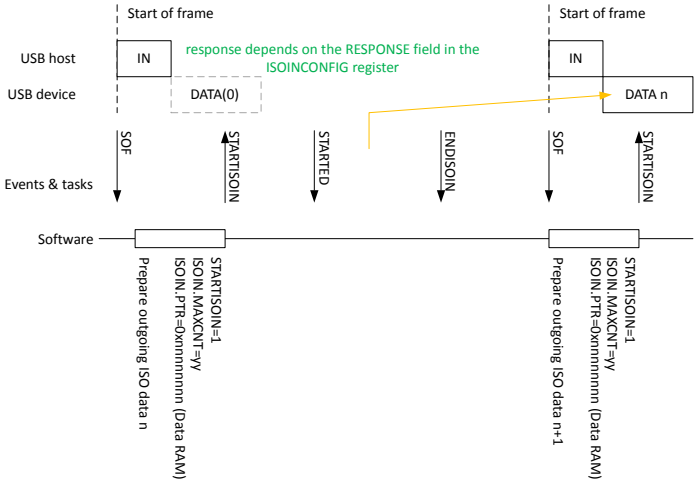

Figure 204: Isochronous IN transfer

6.35.11.2 Isochronous OUT transaction

When the host wants to send isochronous (ISO) data, it issues an OUT token on the isochronous endpoint, followed by data.

The ISO OUT data endpoint has to be explicitly enabled by software through the ISOOUT0 bit in register EPOUTEN.

The amount of last received ISO OUT data is provided in the SIZE.ISOOUT register. Software shall interpret the ZERO and SIZE fields as follows:

ZERO	SIZE	Last received data size
Normal	0	No data received at all
Normal	1..1023	1..1023 bytes of data received
ZeroData	(not of interest)	Zero-length data packet received

Table 132: ISO OUT incoming data size

When EasyDMA is prepared and started, triggering a STARTISOOUT task initiates an EasyDMA transfer to Data RAM. Software shall synchronize ISO OUT transfers with the SOF events. If OUT data is not consumed and processed until next SOF, it will be overwritten by more recent data. EasyDMA uses the address in ISOOUT.PTR and size in ISOOUT.MAXCNT for every new transfer.

Note: The maximum size of an isochronous OUT transfer in USB full-speed is 1023 bytes, and the data buffer in RAM has to be a multiple of 4 bytes and 32-bit aligned in Data RAM. However, the amount of bytes transferred on the USB data endpoint can be of any size (up to 1023 bytes if not shared with an IN ISO endpoint).

If the last received ISO data packet is corrupted (wrong CRC), the USB controller generates an USBEVENT event (at the same time as SOF) and indicates a CRC error on ISOOUTCRC in register EVENTCAUSE. EasyDMA will transfer the data anyway if it has been set up properly.

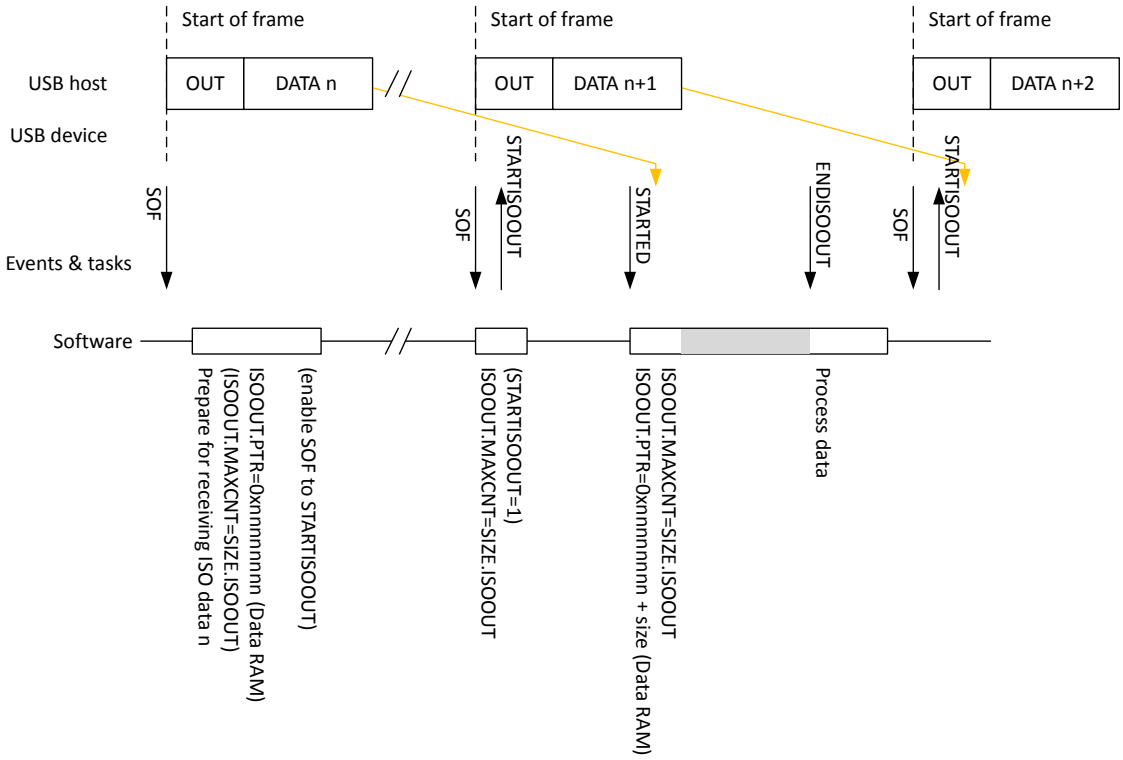

Figure 205: Isochronous OUT transfer

6.35.12 USB register access limitations

Some of the registers in USBD cannot be accessed in specific conditions.

This may be the case when USBD is not enabled (using the ENABLE register) and ready (signalled by the READY bit in EVENTCAUSE after a USBEVENT event), or when USBD is in low power mode while the USB bus is suspended.

Triggering any tasks, including the tasks triggered through the PPI, is affected by this behavior. In addition, the following registers are affected:

- HALTED.EPIN[0..7]
- HALTED.EPOUT[0..7]
- USBADDR
- BMREQUESTTYPE
- BREQUEST
- WVALUEL
- WVALUEH
- WINDEXL
- WINDEXH
- WLENGTHL
- WLENGTHH
- SIZE.EPOUT[0..7]
- SIZE.ISOOUT
- USBPULLUP
- DTOGGLE

- EPINEN
- EPOUTEN
- EPSTALL
- ISOSPLIT
- FRAMECNTR

6.35.13 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40027000	USBD	USBD	Universal serial bus device	

Table 133: Instances

Register	Offset	Description
TASKS_STARTEPIN[0]	0x004	Captures the EPIN[0].PTR and EPIN[0].MAXCNT registers values, and enables endpoint IN 0 to respond to traffic from host
TASKS_STARTEPIN[1]	0x008	Captures the EPIN[1].PTR and EPIN[1].MAXCNT registers values, and enables endpoint IN 1 to respond to traffic from host
TASKS_STARTEPIN[2]	0x00C	Captures the EPIN[2].PTR and EPIN[2].MAXCNT registers values, and enables endpoint IN 2 to respond to traffic from host
TASKS_STARTEPIN[3]	0x010	Captures the EPIN[3].PTR and EPIN[3].MAXCNT registers values, and enables endpoint IN 3 to respond to traffic from host
TASKS_STARTEPIN[4]	0x014	Captures the EPIN[4].PTR and EPIN[4].MAXCNT registers values, and enables endpoint IN 4 to respond to traffic from host
TASKS_STARTEPIN[5]	0x018	Captures the EPIN[5].PTR and EPIN[5].MAXCNT registers values, and enables endpoint IN 5 to respond to traffic from host
TASKS_STARTEPIN[6]	0x01C	Captures the EPIN[6].PTR and EPIN[6].MAXCNT registers values, and enables endpoint IN 6 to respond to traffic from host
TASKS_STARTEPIN[7]	0x020	Captures the EPIN[7].PTR and EPIN[7].MAXCNT registers values, and enables endpoint IN 7 to respond to traffic from host
TASKS_STARTISOIN	0x024	Captures the ISOIN.PTR and ISOIN.MAXCNT registers values, and enables sending data on ISO endpoint
TASKS_STARTEPOUT[0]	0x028	Captures the EPOUT[0].PTR and EPOUT[0].MAXCNT registers values, and enables endpoint 0 to respond to traffic from host
TASKS_STARTEPOUT[1]	0x02C	Captures the EPOUT[1].PTR and EPOUT[1].MAXCNT registers values, and enables endpoint 1 to respond to traffic from host
TASKS_STARTEPOUT[2]	0x030	Captures the EPOUT[2].PTR and EPOUT[2].MAXCNT registers values, and enables endpoint 2 to respond to traffic from host
TASKS_STARTEPOUT[3]	0x034	Captures the EPOUT[3].PTR and EPOUT[3].MAXCNT registers values, and enables endpoint 3 to respond to traffic from host
TASKS_STARTEPOUT[4]	0x038	Captures the EPOUT[4].PTR and EPOUT[4].MAXCNT registers values, and enables endpoint 4 to respond to traffic from host
TASKS_STARTEPOUT[5]	0x03C	Captures the EPOUT[5].PTR and EPOUT[5].MAXCNT registers values, and enables endpoint 5 to respond to traffic from host
TASKS_STARTEPOUT[6]	0x040	Captures the EPOUT[6].PTR and EPOUT[6].MAXCNT registers values, and enables endpoint 6 to respond to traffic from host
TASKS_STARTEPOUT[7]	0x044	Captures the EPOUT[7].PTR and EPOUT[7].MAXCNT registers values, and enables endpoint 7 to respond to traffic from host
TASKS_STARTISOOUT	0x048	Captures the ISOOUT.PTR and ISOOUT.MAXCNT registers values, and enables receiving of data on ISO endpoint
TASKS_EP0RCVOUT	0x04C	Allows OUT data stage on control endpoint 0
TASKS_EP0STATUS	0x050	Allows status stage on control endpoint 0
TASKS_EP0STALL	0x054	Stalls data and status stage on control endpoint 0

Register	Offset	Description
TASKS_DPDMDRIVE	0x058	Forces D+ and D- lines into the state defined in the DPDMVALUE register
TASKS_DPDMNODRIVE	0x05C	Stops forcing D+ and D- lines into any state (USB engine takes control)
EVENTS_USBRESET	0x100	Signals that a USB reset condition has been detected on USB lines
EVENTS_STARTED	0x104	Confirms that the EPIN[n].PTR and EPIN[n].MAXCNT, or EPOUT[n].PTR and EPOUT[n].MAXCNT registers have been captured on all endpoints reported in the EPSTATUS register
EVENTS_ENDEPIN[0]	0x108	The whole EPIN[0] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[1]	0x10C	The whole EPIN[1] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[2]	0x110	The whole EPIN[2] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[3]	0x114	The whole EPIN[3] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[4]	0x118	The whole EPIN[4] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[5]	0x11C	The whole EPIN[5] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[6]	0x120	The whole EPIN[6] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPIN[7]	0x124	The whole EPIN[7] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_EP0DATADONE	0x128	An acknowledged data transfer has taken place on the control endpoint
EVENTS_ENDISOIN	0x12C	The whole ISOIN buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[0]	0x130	The whole EPOUT[0] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[1]	0x134	The whole EPOUT[1] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[2]	0x138	The whole EPOUT[2] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[3]	0x13C	The whole EPOUT[3] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[4]	0x140	The whole EPOUT[4] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[5]	0x144	The whole EPOUT[5] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[6]	0x148	The whole EPOUT[6] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDEPOUT[7]	0x14C	The whole EPOUT[7] buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_ENDISOOUT	0x150	The whole ISOOUT buffer has been consumed. The RAM buffer can be accessed safely by software.
EVENTS_SOF	0x154	Signals that a SOF (start of frame) condition has been detected on USB lines
EVENTS_USBEVENT	0x158	An event or an error not covered by specific events has occurred. Check EVENTCAUSE register to find the cause.
EVENTS_EP0SETUP	0x15C	A valid SETUP token has been received (and acknowledged) on the control endpoint
EVENTS_EPDATA	0x160	A data transfer has occurred on a data endpoint, indicated by the EPDATASTATUS register
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
EVENTCAUSE	0x400	Details on what caused the USBEVENT event
HALTED.EPIN[0]	0x420	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.

Register	Offset	Description
HALTED.EPIN[1]	0x424	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[2]	0x428	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[3]	0x42C	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[4]	0x430	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[5]	0x434	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[6]	0x438	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPIN[7]	0x43C	IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[0]	0x444	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[1]	0x448	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[2]	0x44C	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[3]	0x450	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[4]	0x454	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[5]	0x458	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[6]	0x45C	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[7]	0x460	OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
EPSTATUS	0x468	Provides information on which endpoint's EasyDMA registers have been captured
EPDATASTATUS	0x46C	Provides information on which endpoint(s) an acknowledged data transfer has occurred (EPDATA event)
USBADDR	0x470	Device USB address
BMREQUESTTYPE	0x480	SETUP data, byte 0, bmRequestType
BREQUEST	0x484	SETUP data, byte 1, bRequest
WVALUEL	0x488	SETUP data, byte 2, LSB of wValue
WVALUEH	0x48C	SETUP data, byte 3, MSB of wValue
WINDEXL	0x490	SETUP data, byte 4, LSB of wIndex
WINDEXH	0x494	SETUP data, byte 5, MSB of wIndex
WLENGTHL	0x498	SETUP data, byte 6, LSB of wLength
WLENGTHH	0x49C	SETUP data, byte 7, MSB of wLength
SIZE.EPOUT[0]	0x4A0	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[1]	0x4A4	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[2]	0x4A8	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[3]	0x4AC	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[4]	0x4B0	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[5]	0x4B4	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[6]	0x4B8	Number of bytes received last in the data stage of this OUT endpoint
SIZE.EPOUT[7]	0x4BC	Number of bytes received last in the data stage of this OUT endpoint
SIZE.ISOOUT	0x4C0	Number of bytes received last on this ISO OUT data endpoint
ENABLE	0x500	Enable USB
USBPULLUP	0x504	Control of the USB pull-up
DPDMVALUE	0x508	State D+ and D- lines will be forced into by the DPDMDRIVE task. The DPDMNODRIVE task reverts the control of the lines to MAC IP (no forcing).
DTOGGLE	0x50C	Data toggle control and status
EPINEN	0x510	Endpoint IN enable
EPOUTEN	0x514	Endpoint OUT enable
EPSTALL	0x518	STALL endpoints
ISOSPLIT	0x51C	Controls the split of ISO buffers

Register	Offset	Description
FRAMECNTR	0x520	Returns the current value of the start of frame counter
LOWPOWER	0x52C	Controls USBD peripheral low power mode during USB suspend
ISOINCONFIG	0x530	Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent
EPIN[0].PTR	0x600	Data pointer
EPIN[0].MAXCNT	0x604	Maximum number of bytes to transfer
EPIN[0].AMOUNT	0x608	Number of bytes transferred in the last transaction
EPIN[1].PTR	0x614	Data pointer
EPIN[1].MAXCNT	0x618	Maximum number of bytes to transfer
EPIN[1].AMOUNT	0x61C	Number of bytes transferred in the last transaction
EPIN[2].PTR	0x628	Data pointer
EPIN[2].MAXCNT	0x62C	Maximum number of bytes to transfer
EPIN[2].AMOUNT	0x630	Number of bytes transferred in the last transaction
EPIN[3].PTR	0x63C	Data pointer
EPIN[3].MAXCNT	0x640	Maximum number of bytes to transfer
EPIN[3].AMOUNT	0x644	Number of bytes transferred in the last transaction
EPIN[4].PTR	0x650	Data pointer
EPIN[4].MAXCNT	0x654	Maximum number of bytes to transfer
EPIN[4].AMOUNT	0x658	Number of bytes transferred in the last transaction
EPIN[5].PTR	0x664	Data pointer
EPIN[5].MAXCNT	0x668	Maximum number of bytes to transfer
EPIN[5].AMOUNT	0x66C	Number of bytes transferred in the last transaction
EPIN[6].PTR	0x678	Data pointer
EPIN[6].MAXCNT	0x67C	Maximum number of bytes to transfer
EPIN[6].AMOUNT	0x680	Number of bytes transferred in the last transaction
EPIN[7].PTR	0x68C	Data pointer
EPIN[7].MAXCNT	0x690	Maximum number of bytes to transfer
EPIN[7].AMOUNT	0x694	Number of bytes transferred in the last transaction
ISOIN.PTR	0x6A0	Data pointer
ISOIN.MAXCNT	0x6A4	Maximum number of bytes to transfer
ISOIN.AMOUNT	0x6A8	Number of bytes transferred in the last transaction
EPOUT[0].PTR	0x700	Data pointer
EPOUT[0].MAXCNT	0x704	Maximum number of bytes to transfer
EPOUT[0].AMOUNT	0x708	Number of bytes transferred in the last transaction
EPOUT[1].PTR	0x714	Data pointer
EPOUT[1].MAXCNT	0x718	Maximum number of bytes to transfer
EPOUT[1].AMOUNT	0x71C	Number of bytes transferred in the last transaction
EPOUT[2].PTR	0x728	Data pointer
EPOUT[2].MAXCNT	0x72C	Maximum number of bytes to transfer
EPOUT[2].AMOUNT	0x730	Number of bytes transferred in the last transaction
EPOUT[3].PTR	0x73C	Data pointer
EPOUT[3].MAXCNT	0x740	Maximum number of bytes to transfer
EPOUT[3].AMOUNT	0x744	Number of bytes transferred in the last transaction
EPOUT[4].PTR	0x750	Data pointer
EPOUT[4].MAXCNT	0x754	Maximum number of bytes to transfer
EPOUT[4].AMOUNT	0x758	Number of bytes transferred in the last transaction
EPOUT[5].PTR	0x764	Data pointer
EPOUT[5].MAXCNT	0x768	Maximum number of bytes to transfer
EPOUT[5].AMOUNT	0x76C	Number of bytes transferred in the last transaction
EPOUT[6].PTR	0x778	Data pointer
EPOUT[6].MAXCNT	0x77C	Maximum number of bytes to transfer
EPOUT[6].AMOUNT	0x780	Number of bytes transferred in the last transaction
EPOUT[7].PTR	0x78C	Data pointer
EPOUT[7].MAXCNT	0x790	Maximum number of bytes to transfer

Register	Offset	Description
EPOUT[7].AMOUNT	0x794	Number of bytes transferred in the last transaction
ISOOUT.PTR	0x7A0	Data pointer
ISOOUT.MAXCNT	0x7A4	Maximum number of bytes to transfer
ISOOUT.AMOUNT	0x7A8	Number of bytes transferred in the last transaction

Table 134: Register overview

6.35.13.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EP0DATADONE_STARTEPIN0			Shortcut between EP0DATADONE event and STARTEPIN[0] task See EVENTS_EP0DATADONE and TASKS_STARTEPIN[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	EP0DATADONE_STARTEP			Shortcut between EP0DATADONE event and STARTEPOUT[0] task See EVENTS_EP0DATADONE and TASKS_STARTEPOUT[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	EP0DATADONE_EP0STATUS			Shortcut between EP0DATADONE event and EP0STATUS task See EVENTS_EP0DATADONE and TASKS_EP0STATUS
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDEPOUT0_EP0STATUS			Shortcut between ENDEPOUT[0] event and EP0STATUS task See EVENTS_ENDEPOUT[0] and TASKS_EP0STATUS
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	ENDEPOUT0_EP0RCVOUT			Shortcut between ENDEPOUT[0] event and EP0RCVOUT task See EVENTS_ENDEPOUT[0] and TASKS_EP0RCVOUT
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.35.13.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	USBRESET			Enable or disable interrupt for USBRESET event See EVENTS_USBRESET

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	STARTED			Enable or disable interrupt for STARTED event See EVENTS_STARTED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	ENDEPIN0			Enable or disable interrupt for ENDEPIN[0] event See EVENTS_ENDEPIN[0]
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ENDEPIN1			Enable or disable interrupt for ENDEPIN[1] event See EVENTS_ENDEPIN[1]
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	ENDEPIN2			Enable or disable interrupt for ENDEPIN[2] event See EVENTS_ENDEPIN[2]
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	ENDEPIN3			Enable or disable interrupt for ENDEPIN[3] event See EVENTS_ENDEPIN[3]
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	ENDEPIN4			Enable or disable interrupt for ENDEPIN[4] event See EVENTS_ENDEPIN[4]
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	ENDEPIN5			Enable or disable interrupt for ENDEPIN[5] event See EVENTS_ENDEPIN[5]
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	ENDEPIN6			Enable or disable interrupt for ENDEPIN[6] event See EVENTS_ENDEPIN[6]
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	ENDEPIN7			Enable or disable interrupt for ENDEPIN[7] event See EVENTS_ENDEPIN[7]
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	EP0DATADONE			Enable or disable interrupt for EP0DATADONE event See EVENTS_EP0DATADONE
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	ENDISOIN			Enable or disable interrupt for ENDISOIN event See EVENTS_ENDISOIN
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
M	RW	ENDEPOUT0			Enable or disable interrupt for ENDEPOUT[0] event See EVENTS_ENDEPOUT[0]
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	ENDEPOUT1			Enable or disable interrupt for ENDEPOUT[1] event See EVENTS_ENDEPOUT[1]
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	ENDEPOUT2			Enable or disable interrupt for ENDEPOUT[2] event See EVENTS_ENDEPOUT[2]
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	ENDEPOUT3			Enable or disable interrupt for ENDEPOUT[3] event See EVENTS_ENDEPOUT[3]
			Disabled	0	Disable
			Enabled	1	Enable
Q	RW	ENDEPOUT4			Enable or disable interrupt for ENDEPOUT[4] event See EVENTS_ENDEPOUT[4]
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	ENDEPOUT5			Enable or disable interrupt for ENDEPOUT[5] event See EVENTS_ENDEPOUT[5]
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	ENDEPOUT6			Enable or disable interrupt for ENDEPOUT[6] event See EVENTS_ENDEPOUT[6]
			Disabled	0	Disable
			Enabled	1	Enable
T	RW	ENDEPOUT7			Enable or disable interrupt for ENDEPOUT[7] event See EVENTS_ENDEPOUT[7]
			Disabled	0	Disable
			Enabled	1	Enable
U	RW	ENDISOOUT			Enable or disable interrupt for ENDISOOUT event See EVENTS_ENDISOOUT
			Disabled	0	Disable
			Enabled	1	Enable
V	RW	SOF			Enable or disable interrupt for SOF event See EVENTS_SOF
			Disabled	0	Disable
			Enabled	1	Enable
W	RW	USBEVENT			Enable or disable interrupt for USBEVENT event See EVENTS_USBEVENT
			Disabled	0	Disable
			Enabled	1	Enable
X	RW	EP0SETUP			Enable or disable interrupt for EP0SETUP event See EVENTS_EP0SETUP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
Y	RW	EPDATA			Enable or disable interrupt for EPDATA event See EVENTS_EPDATA
			Disabled	0	Disable
			Enabled	1	Enable

6.35.13.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	USBRESET			Write '1' to enable interrupt for USBRESET event See EVENTS_USBRESET
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STARTED			Write '1' to enable interrupt for STARTED event See EVENTS_STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ENDEPIN0			Write '1' to enable interrupt for ENDEPIN[0] event See EVENTS_ENDEPIN[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDEPIN1			Write '1' to enable interrupt for ENDEPIN[1] event See EVENTS_ENDEPIN[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	ENDEPIN2			Write '1' to enable interrupt for ENDEPIN[2] event See EVENTS_ENDEPIN[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDEPIN3			Write '1' to enable interrupt for ENDEPIN[3] event See EVENTS_ENDEPIN[3]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
G	RW	ENDEPIN4			Write '1' to enable interrupt for ENDEPIN[4] event See EVENTS_ENDEPIN[4]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	ENDEPIN5			Write '1' to enable interrupt for ENDEPIN[5] event See EVENTS_ENDEPIN[5]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	ENDEPIN6			Write '1' to enable interrupt for ENDEPIN[6] event See EVENTS_ENDEPIN[6]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	ENDEPIN7			Write '1' to enable interrupt for ENDEPIN[7] event See EVENTS_ENDEPIN[7]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	EP0DATADONE			Write '1' to enable interrupt for EP0DATADONE event See EVENTS_EP0DATADONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDISOIN			Write '1' to enable interrupt for ENDISOIN event See EVENTS_ENDISOIN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDEPOUT0			Write '1' to enable interrupt for ENDEPOUT[0] event See EVENTS_ENDEPOUT[0]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	ENDEPOUT1			Write '1' to enable interrupt for ENDEPOUT[1] event See EVENTS_ENDEPOUT[1]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	ENDEPOUT2			Write '1' to enable interrupt for ENDEPOUT[2] event See EVENTS_ENDEPOUT[2]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	ENDEPOUT3			Write '1' to enable interrupt for ENDEPOUT[3] event See EVENTS_ENDEPOUT[3]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	ENDEPOUT4			Write '1' to enable interrupt for ENDEPOUT[4] event See EVENTS_ENDEPOUT[4]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	ENDEPOUT5			Write '1' to enable interrupt for ENDEPOUT[5] event See EVENTS_ENDEPOUT[5]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	ENDEPOUT6			Write '1' to enable interrupt for ENDEPOUT[6] event See EVENTS_ENDEPOUT[6]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	ENDEPOUT7			Write '1' to enable interrupt for ENDEPOUT[7] event See EVENTS_ENDEPOUT[7]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	ENDISOOUT			Write '1' to enable interrupt for ENDISOOUT event See EVENTS_ENDISOOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	SOF			Write '1' to enable interrupt for SOF event See EVENTS_SOF
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
W	RW	USBEVENT			Write '1' to enable interrupt for USBEVENT event See EVENTS_USBEVENT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
X	RW	EP0SETUP			Write '1' to enable interrupt for EP0SETUP event See EVENTS_EP0SETUP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Y	RW	EPDATA			Write '1' to enable interrupt for EPDATA event See EVENTS_EPDATA
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled

6.35.13.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	USBRESET			Write '1' to disable interrupt for USBRESET event See EVENTS_USBRESET
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STARTED			Write '1' to disable interrupt for STARTED event See EVENTS_STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ENDEPIN0			Write '1' to disable interrupt for ENDEPIN[0] event See EVENTS_ENDEPIN[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDEPIN1			Write '1' to disable interrupt for ENDEPIN[1] event See EVENTS_ENDEPIN[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	ENDEPIN2			Write '1' to disable interrupt for ENDEPIN[2] event See EVENTS_ENDEPIN[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDEPIN3			Write '1' to disable interrupt for ENDEPIN[3] event See EVENTS_ENDEPIN[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ENDEPIN4			Write '1' to disable interrupt for ENDEPIN[4] event See EVENTS_ENDEPIN[4]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
H	RW	ENDEPIN5			Write '1' to disable interrupt for ENDEPIN[5] event See EVENTS_ENDEPIN[5]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	ENDEPIN6			Write '1' to disable interrupt for ENDEPIN[6] event See EVENTS_ENDEPIN[6]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	ENDEPIN7			Write '1' to disable interrupt for ENDEPIN[7] event See EVENTS_ENDEPIN[7]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	EP0DATADONE			Write '1' to disable interrupt for EP0DATADONE event See EVENTS_EP0DATADONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDISOIN			Write '1' to disable interrupt for ENDISOIN event See EVENTS_ENDISOIN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDEPOUT0			Write '1' to disable interrupt for ENDEPOUT[0] event See EVENTS_ENDEPOUT[0]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	ENDEPOUT1			Write '1' to disable interrupt for ENDEPOUT[1] event See EVENTS_ENDEPOUT[1]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	ENDEPOUT2			Write '1' to disable interrupt for ENDEPOUT[2] event See EVENTS_ENDEPOUT[2]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	ENDEPOUT3			Write '1' to disable interrupt for ENDEPOUT[3] event See EVENTS_ENDEPOUT[3]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	ENDEPOUT4			Write '1' to disable interrupt for ENDEPOUT[4] event See EVENTS_ENDEPOUT[4]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	ENDEPOUT5			Write '1' to disable interrupt for ENDEPOUT[5] event See EVENTS_ENDEPOUT[5]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	ENDEPOUT6			Write '1' to disable interrupt for ENDEPOUT[6] event See EVENTS_ENDEPOUT[6]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	ENDEPOUT7			Write '1' to disable interrupt for ENDEPOUT[7] event See EVENTS_ENDEPOUT[7]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	ENDISOOUT			Write '1' to disable interrupt for ENDISOOUT event See EVENTS_ENDISOOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	SOF			Write '1' to disable interrupt for SOF event See EVENTS_SOF
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
W	RW	USBEVENT			Write '1' to disable interrupt for USBEVENT event See EVENTS_USBEVENT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
X	RW	EP0SETUP			Write '1' to disable interrupt for EP0SETUP event See EVENTS_EP0SETUP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Y	RW	EPDATA			Write '1' to disable interrupt for EPDATA event See EVENTS_EPDATA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.35.13.5 EVENTCAUSE

Address offset: 0x400

Details on what caused the USBEVENT event

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ISOOUTCRC			CRC error was detected on isochronous OUT endpoint 8. Write '1' to clear.
			NotDetected	0	No error detected
			Detected	1	Error detected
B	RW	SUSPEND			Signals that USB lines have been idle long enough for the device to enter suspend. Write '1' to clear.
			NotDetected	0	Suspend not detected
			Detected	1	Suspend detected
C	RW	RESUME			Signals that a RESUME condition (K state or activity restart) has been detected on USB lines. Write '1' to clear.
			NotDetected	0	Resume not detected
			Detected	1	Resume detected
D	RW	USBWUALLOWED			USB MAC has been woken up and operational. Write '1' to clear.
			NotAllowed	0	Wake up not allowed
			Allowed	1	Wake up allowed
E	RW	READY			USB device is ready for normal operation. Write '1' to clear.
			NotDetected	0	USBEVENT was not issued due to USBD peripheral ready
			Ready	1	USBD peripheral is ready

6.35.13.6 HALTED.EPIN[n] (n=0..7)

Address offset: 0x420 + (n × 0x4)

IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	GETSTATUS			IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
			NotHalted	0	Endpoint is not halted
			Halted	1	Endpoint is halted

6.35.13.7 HALTED.EPOUT[n] (n=0..7)

Address offset: 0x444 + (n × 0x4)

OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	GETSTATUS			OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
			NotHalted	0	Endpoint is not halted
			Halted	1	Endpoint is halted

6.35.13.8 EPSTATUS

Address offset: 0x468

Provides information on which endpoint's EasyDMA registers have been captured

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-I	RW	EPIN[i] (i=0..8)			Captured state of endpoint's EasyDMA registers. Write '1' to clear.
			NoData	0	EasyDMA registers have not been captured for this endpoint
			DataDone	1	EasyDMA registers have been captured for this endpoint
J-R	RW	EPOUT[i] (i=0..8)			Captured state of endpoint's EasyDMA registers. Write '1' to clear.
			NoData	0	EasyDMA registers have not been captured for this endpoint
			DataDone	1	EasyDMA registers have been captured for this endpoint

6.35.13.9 EPDATASTATUS

Address offset: 0x46C

Provides information on which endpoint(s) an acknowledged data transfer has occurred (EPDATA event)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A-G	RW	EPIN[i] (i=1..7)			Acknowledged data transfer on this IN endpoint. Write '1' to clear.
			NotDone	0	No acknowledged data transfer on this endpoint
			DataDone	1	Acknowledged data transfer on this endpoint has occurred
H-N	RW	EPOUT[i] (i=1..7)			Acknowledged data transfer on this OUT endpoint. Write '1' to clear.
			NotStarted	0	No acknowledged data transfer on this endpoint
			Started	1	Acknowledged data transfer on this endpoint has occurred

6.35.13.10 USBADDR

Address offset: 0x470

Device USB address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	ADDR			Device USB address

6.35.13.11 BMREQUESTTYPE

Address offset: 0x480

SETUP data, byte 0, bmRequestType

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RECIPIENT			Data transfer type
			Device	0	Device
			Interface	1	Interface
			Endpoint	2	Endpoint
			Other	3	Other
B	R	TYPE			Data transfer type
			Standard	0	Standard
			Class	1	Class
			Vendor	2	Vendor
C	R	DIRECTION			Data transfer direction
			HostToDevice	0	Host-to-device
			DeviceToHost	1	Device-to-host

6.35.13.12 BREQUEST

Address offset: 0x484

SETUP data, byte 1, bRequest

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	BREQUEST			SETUP data, byte 1, bRequest. Values provided for standard requests only, user must implement class and vendor values.
			STD_GET_STATUS	0	Standard request GET_STATUS
			STD_CLEAR_FEATURE	1	Standard request CLEAR_FEATURE
			STD_SET_FEATURE	3	Standard request SET_FEATURE
			STD_SET_ADDRESS	5	Standard request SET_ADDRESS
			STD_GET_DESCRIPTOR	6	Standard request GET_DESCRIPTOR
			STD_SET_DESCRIPTOR	7	Standard request SET_DESCRIPTOR
			STD_GET_CONFIGURATION	8	Standard request GET_CONFIGURATION
			STD_SET_CONFIGURATION	9	Standard request SET_CONFIGURATION
			STD_GET_INTERFACE	10	Standard request GET_INTERFACE
			STD_SET_INTERFACE	11	Standard request SET_INTERFACE
			STD_SYNCH_FRAME	12	Standard request SYNCH_FRAME

6.35.13.13 WVALUEL

Address offset: 0x488

SETUP data, byte 2, LSB of wValue

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	WVALUEL			SETUP data, byte 2, LSB of wValue

6.35.13.14 WVALUEH

Address offset: 0x48C

SETUP data, byte 3, MSB of wValue

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	WVALUEH			SETUP data, byte 3, MSB of wValue

6.35.13.15 WINDEXL

Address offset: 0x490

SETUP data, byte 4, LSB of wIndex

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	WINDEXL			SETUP data, byte 4, LSB of wIndex

6.35.13.16 WINDEXH

Address offset: 0x494

SETUP data, byte 5, MSB of wIndex

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	WINDEXH			SETUP data, byte 5, MSB of wIndex

6.35.13.17 WLENGTHL

Address offset: 0x498

SETUP data, byte 6, LSB of wLength

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	R	WLENGTHL			SETUP data, byte 6, LSB of wLength

6.35.13.18 WLENGTHH

Address offset: 0x49C

SETUP data, byte 7, MSB of wLength

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	WLENGTHH			SETUP data, byte 7, MSB of wLength

6.35.13.19 SIZE.EPOUT[n] (n=0..7)

Address offset: 0x4A0 + (n × 0x4)

Number of bytes received last in the data stage of this OUT endpoint

Write to any value to accept further OUT traffic on this endpoint, and overwrite the intermediate buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SIZE			Number of bytes received last in the data stage of this OUT endpoint

6.35.13.20 SIZE.ISOOUT

Address offset: 0x4C0

Number of bytes received last on this ISO OUT data endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00010000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	SIZE			Number of bytes received last on this ISO OUT data endpoint
B	R	ZERO			Zero-length data packet received
			Normal	0	No zero-length data received, use value in SIZE
			ZeroData	1	Zero-length data received, ignore value in SIZE

6.35.13.21 ENABLE

Address offset: 0x500

Enable USB

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	ENABLE			Enable USB
			Disabled	0	USB peripheral is disabled
			Enabled	1	USB peripheral is enabled

6.35.13.22 USBPULLUP

Address offset: 0x504

Control of the USB pull-up

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	CONNECT			Control of the USB pull-up on the D+ line
			Disabled	0	Pull-up is disconnected
			Enabled	1	Pull-up is connected to D+

6.35.13.23 DPDMVALUE

Address offset: 0x508

State D+ and D- lines will be forced into by the DPDMDRIVE task. The DPDMNODRIVE task reverts the control of the lines to MAC IP (no forcing).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	STATE			State D+ and D- lines will be forced into by the DPDMDRIVE task
			Resume	1	D+ forced low, D- forced high (K state) for a timing preset in hardware (50 µs or 5 ms, depending on bus state)
			J	2	D+ forced high, D- forced low (J state)
			K	4	D+ forced low, D- forced high (K state)

6.35.13.24 DTOGGLE

Address offset: 0x50C

Data toggle control and status

Write this register first with VALUE=Nop to select the endpoint; then read it to get the status from VALUE, or write it again with VALUE=Data0 or Data1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C B A A A
Reset 0x00000100				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	EP			Select bulk endpoint number
B	RW	IO			Selects IN or OUT endpoint
			Out	0	Selects OUT endpoint
			In	1	Selects IN endpoint
C	RW	VALUE			Data toggle value
			Nop	0	No action on data toggle when writing the register with this value
			Data0	1	Data toggle is DATA0 on endpoint set by EP and IO
			Data1	2	Data toggle is DATA1 on endpoint set by EP and IO

6.35.13.25 EPINEN

Address offset: 0x510

Endpoint IN enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A-H	RW	IN[i] (i=0..7)			Enable IN endpoint i
			Disable	0	Disable endpoint IN i (no response to IN tokens)
			Enable	1	Enable endpoint IN i (response to IN tokens)
I	RW	ISOIN			Enable ISO IN endpoint
			Disable	0	Disable ISO IN endpoint 8
			Enable	1	Enable ISO IN endpoint 8

6.35.13.26 EPOUTEN

Address offset: 0x514

Endpoint OUT enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A-H	RW	OUT[i] (i=0..7)			Enable OUT endpoint i
			Disable	0	Disable endpoint OUT i (no response to OUT tokens)
			Enable	1	Enable endpoint OUT i (response to OUT tokens)
I	RW	ISOOUT			Enable ISO OUT endpoint 8
			Disable	0	Disable ISO OUT endpoint 8
			Enable	1	Enable ISO OUT endpoint 8

6.35.13.27 EPSTALL

Address offset: 0x518

STALL endpoints

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	W	EP			Select endpoint number
B	W	IO			Selects IN or OUT endpoint
			Out	0	Selects OUT endpoint
			In	1	Selects IN endpoint
C	W	STALL			Stall selected endpoint
			UnStall	0	Don't stall selected endpoint
			Stall	1	Stall selected endpoint

6.35.13.28 ISOSPLIT

Address offset: 0x51C

Controls the split of ISO buffers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	SPLIT			Controls the split of ISO buffers
			OneDir	0x0000	Full buffer dedicated to either iso IN or OUT
			HalfIN	0x0080	Lower half for IN, upper half for OUT

6.35.13.29 FRAMECNTR

Address offset: 0x520

Returns the current value of the start of frame counter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	FRAMECNTR			Returns the current value of the start of frame counter

6.35.13.30 LOWPOWER

Address offset: 0x52C

Controls USBD peripheral low power mode during USB suspend

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	LOWPOWER			Controls USBD peripheral low-power mode during USB suspend
			ForceNormal	0	Software must write this value to exit low power mode and before performing a remote wake-up
			LowPower	1	Software must write this value to enter low power mode after DMA and software have finished interacting with the USB peripheral

6.35.13.31 ISOINCONFIG

Address offset: 0x530

Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	RESPONSE			Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent
			NoResp	0	Endpoint does not respond in that case
			ZeroData	1	Endpoint responds with a zero-length data packet in that case

6.35.13.32 EPIN[n].PTR (n=0..7)

Address offset: 0x600 + (n × 0x14)

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer. Accepts any address in Data RAM.

6.35.13.33 EPIN[n].MAXCNT (n=0..7)

Address offset: 0x604 + (n × 0x14)

Maximum number of bytes to transfer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[64..0]	Maximum number of bytes to transfer

6.35.13.34 EPIN[n].AMOUNT (n=0..7)

Address offset: 0x608 + (n × 0x14)

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

6.35.13.35 ISOIN.PTR

Address offset: 0x6A0

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer. Accepts any address in Data RAM.

6.35.13.36 ISOIN.MAXCNT

Address offset: 0x6A4

Maximum number of bytes to transfer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[1023..1]	Maximum number of bytes to transfer

6.35.13.37 ISOIN.AMOUNT

Address offset: 0x6A8

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

6.35.13.38 EPOUT[n].PTR (n=0..7)

Address offset: 0x700 + (n × 0x14)

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer. Accepts any address in Data RAM.

6.35.13.39 EPOUT[n].MAXCNT (n=0..7)

Address offset: 0x704 + (n × 0x14)

Maximum number of bytes to transfer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT		[64..0]	Maximum number of bytes to transfer

6.35.13.40 EPOUT[n].AMOUNT (n=0..7)

Address offset: 0x708 + (n × 0x14)

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

6.35.13.41 ISOOUT.PTR

Address offset: 0x7A0

Data pointer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	PTR			Data pointer. Accepts any address in Data RAM.

6.35.13.42 ISOOUT.MAXCNT

Address offset: 0x7A4

Maximum number of bytes to transfer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	RW	MAXCNT			Maximum number of bytes to transfer

6.35.13.43 ISOOUT.AMOUNT

Address offset: 0x7A8

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	RW	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

6.35.14 Electrical specification

6.35.14.1 USB Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,ACTIVE,VBUS}$	Current from VBUS supply, USB active		2.4		mA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,SUSPEND,VBUS}$	Current from VBUS supply, USB suspended, CPU sleeping		262		µA
$I_{USB,ACTIVE,VDD}$	Current from VDD supply (normal voltage mode), 256 kB RAM retention, regulator=LDO, CPU running, USB active		7.73		mA
$I_{USB,SUSPEND,VDD}$	Current from VDD supply (normal voltage mode), 256 kB RAM retention, regulator=LDO, CPU sleeping, USB suspended		173		µA
$I_{USB,ACTIVE,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (REG0 output), 256 kB RAM retention, regulator=LDO, CPU running, USB active		7.46		mA
$I_{USB,SUSPEND,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (REG0 output), 256 kB RAM retention, regulator=LDO, CPU sleeping, USB suspended		178		µA
$I_{USB,DISABLED,VDD}$	Current from VDD supply, USB disabled, VBUS supply connected, 256 kB RAM retention, regulator=LDO, CPU sleeping	3	7	11	µA
$R_{USB,PU,ACTIVE}$	Value of pull-up on D+, bus active (upstream device transmitting)	1425	2300	3090	Ω
$R_{USB,PU,IDLE}$	Value of pull-up on D+, bus idle	900	1200	1575	Ω
$t_{USB,DETRST}$	Minimum duration of an SE0 state to be detected as a USB reset condition				µs
$f_{USB,CLK}$	Frequency of local clock, USB active		48		MHz
$f_{USB,TOL}$	Accuracy of local clock, USB active[42]			±1000	ppm
$T_{USB,JITTER}$	Jitter on USB local clock, USB active			±1	ns

6.36 WDT — Watchdog timer

A countdown watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up.

The watchdog timer is started by triggering the START task.

The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU. The watchdog is implemented as a down-counter that generates a TIMEOUT event when it wraps over after counting down to 0. When the watchdog timer is started through the START task, the watchdog counter is loaded with the value specified in the CRV register. This counter is also reloaded with the value specified in the CRV register when a reload request is granted.

The watchdog's timeout period is given by:

```
timeout [s] = ( CRV + 1 ) / 32768
```

When started, the watchdog will automatically force the 32.768 kHz RC oscillator on as long as no other 32.768 kHz clock source is running and generating the 32.768 kHz system clock, see chapter CLOCK — Clock control on page 80.

6.36.1 Reload criteria

The watchdog has eight separate reload request registers, which shall be used to request the watchdog to reload its counter with the value specified in the CRV register. To reload the watchdog counter, the special value 0x6E524635 needs to be written to all enabled reload registers.

[42] The local clock can be stopped during USB suspend

One or more RR registers can be individually enabled through the RREN register.

6.36.2 Temporarily pausing the watchdog

By default, the watchdog will be active counting down the down-counter while the CPU is sleeping and when it is halted by the debugger. It is however possible to configure the watchdog to automatically pause while the CPU is sleeping as well as when it is halted by the debugger.

6.36.3 Watchdog reset

A TIMEOUT event will automatically lead to a watchdog reset.

See Reset on page 68 for more information about reset sources. If the watchdog is configured to generate an interrupt on the TIMEOUT event, the watchdog reset will be postponed with two 32.768 kHz clock cycles after the TIMEOUT event has been generated. Once the TIMEOUT event has been generated, the impending watchdog reset will always be effectuated.

The watchdog must be configured before it is started. After it is started, the watchdog's configuration registers, which comprise registers CRV, RREN, and CONFIG, will be blocked for further configuration.

The watchdog can be reset from several reset sources, see Reset behavior on page 69.

When the device starts running again, after a reset, or waking up from OFF mode, the watchdog configuration registers will be available for configuration again.

6.36.4 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40010000	WDT	WDT	Watchdog timer	

Table 135: Instances

Register	Offset	Description
TASKS_START	0x000	Start the watchdog
EVENTS_TIMEOUT	0x100	Watchdog timeout
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RUNSTATUS	0x400	Run status
REQSTATUS	0x404	Request status
CRV	0x504	Counter reload value
RREN	0x508	Enable register for reload request registers
CONFIG	0x50C	Configuration register
RR[0]	0x600	Reload request 0
RR[1]	0x604	Reload request 1
RR[2]	0x608	Reload request 2
RR[3]	0x60C	Reload request 3
RR[4]	0x610	Reload request 4
RR[5]	0x614	Reload request 5
RR[6]	0x618	Reload request 6
RR[7]	0x61C	Reload request 7

Table 136: Register overview

6.36.4.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to enable interrupt for TIMEOUT event See EVENTS_TIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.36.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to disable interrupt for TIMEOUT event See EVENTS_TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.36.4.3 RUNSTATUS

Address offset: 0x400

Run status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	RW	Field	Value ID	Value	Description
A	R	RUNSTATUS			Indicates whether or not the watchdog is running
			NotRunning	0	Watchdog not running
			Running	1	Watchdog is running

6.36.4.4 REQSTATUS

Address offset: 0x404

Request status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A-H	R	RR[i] (i=0..7)			Request status for RR[i] register
			DisabledOrRequested	0	RR[i] register is not enabled, or are already requesting reload
			EnabledAndUnrequested	1	RR[i] register is enabled, and are not yet requesting reload

6.36.4.5 CRV

Address offset: 0x504

Counter reload value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	RW	Field	Value ID	Value	Description
A	RW	CRV		[0x0000000F..0xFFFFFFFF]	Counter reload value in number of cycles of the 32.768 kHz clock

6.36.4.6 RREN

Address offset: 0x508

Enable register for reload request registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A-H	RW	RR[i] (i=0..7)			Enable or disable RR[i] register
			Disabled	0	Disable RR[i] register
			Enabled	1	Enable RR[i] register

6.36.4.7 CONFIG

Address offset: 0x50C

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	RW	Field	Value ID	Value	Description
A	RW	SLEEP			Configure the watchdog to either be paused, or kept running, while the CPU is sleeping
			Pause	0	Pause watchdog while the CPU is sleeping
			Run	1	Keep the watchdog running while the CPU is sleeping
C	RW	HALT			Configure the watchdog to either be paused, or kept running, while the CPU is halted by the debugger
			Pause	0	Pause watchdog while the CPU is halted by the debugger
			Run	1	Keep the watchdog running while the CPU is halted by the debugger

6.36.4.8 RR[n] (n=0..7)

Address offset: 0x600 + (n × 0x4)

Reload request n

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	RW	Field	Value ID	Value	Description
A	W	RR			Reload request register
			Reload	0x6E524635	Value to request a reload of the watchdog timer

6.36.5 Electrical specification

6.36.5.1 Watchdog Timer Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{WDT}	Time out interval	458 µs		36 h	

7 Hardware and layout

7.1 Pin assignments

This section describes the pin assignment and the pin functions.

This device provides flexibility when it comes to routing and configuration of the GPIO pins. However, some pins have recommendations for how the pin should be configured or what it should be used for. See aQFN73 ball assignments on page 525 for more information about this.

7.1.1 aQFN73 ball assignments

The ball assignment table and figure describe the assignments for this variant of the chip.

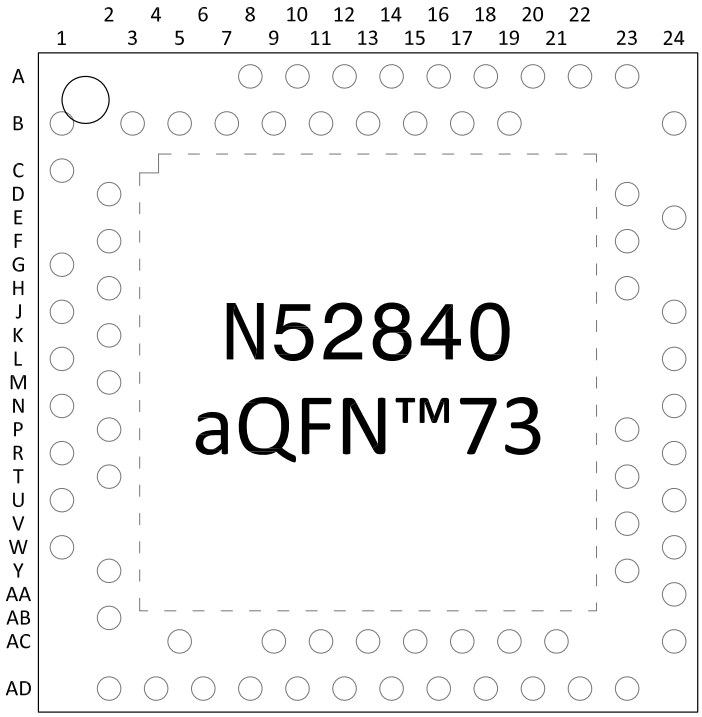

Figure 206: aQFN™73 ball assignments, top view

Pin	Name	Function	Description	Recommended usage
A8	P0.31 AIN7	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
A10	P0.29 AIN5	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
A12	P0.02 AIN0	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
A14	P1.15	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
A16	P1.13	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
A18	DEC2	Power	1.3 V regulator supply decoupling	
A20	P1.10	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
A22	VDD	Power	Power supply	
A23	XC2	Analog input	Connection for 32 MHz crystal	
B1	VDD	Power	Power supply	
B3	DCC	Power	DC/DC converter output	
B5	DEC4	Power	1.3 V regulator supply decoupling. Has to be connected to DEC6 (pin E24).	
B7	VSS	Power	Ground	
B9	P0.30 AIN6	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
B11	P0.28 AIN4	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
B13	P0.03 AIN1	Digital I/O Analog input	General purpose I/O. Analog input.	Standard drive, low frequency I/O only
B15	P1.14	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
B17	P1.12	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
B19	P1.11	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
B24	XC1	Analog input	Connection for 32 MHz crystal	
C1	DEC1	Power	1.1 V regulator supply decoupling	
D2	P0.00 XL1	Digital I/O Analog input	General purpose I/O. Connection for 32.768 kHz crystal.	
D23	DEC3	Power	Power supply, decoupling	
E24	DEC6	Power	1.3 V regulator supply decoupling. Has to be connected to DEC4 (pin B5).	
F2	P0.01 XL2	Digital I/O Analog input	General purpose I/O. Connection for 32.768 kHz crystal.	
F23	VSS_PA	Power	Ground (radio supply)	
G1	P0.26	Digital I/O	General purpose I/O	
H2	P0.27	Digital I/O	General purpose I/O	
H23	ANT	RF	Single-ended radio antenna connection	See Reference circuitry on page 528 for guidelines on how to ensure good RF performance
J1	P0.04 AIN2	Digital I/O Analog input	General purpose I/O. Analog input.	

Pin	Name	Function	Description	Recommended usage
J24	P0.10 NFC2	Digital I/O NFC input	General purpose I/O. NFC antenna connection.	Standard drive, low frequency I/O only
K2	P0.05 AIN3	Digital I/O Analog input	General purpose I/O. Analog input.	
L1	P0.06	Digital I/O	General purpose I/O	
L24	P0.09 NFC1	Digital I/O NFC input	General purpose I/O. NFC antenna connection.	Standard drive, low frequency I/O only
M2	P0.07 TRACECLK	Digital I/O Trace clock	General purpose I/O. Trace buffer clock.	
N1	P0.08	Digital I/O	General purpose I/O	
N24	DEC5	Power	1.3 V regulator supply decoupling	
P2	P1.08	Digital I/O	General purpose I/O	
P23	P1.07	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
R1	P1.09 TRACEDATA3	Digital I/O Trace data	General purpose I/O. Trace buffer TRACEDATA[3].	
R24	P1.06	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
T2	P0.11 TRACEDATA2	Digital I/O Trace data	General purpose I/O. Trace buffer TRACEDATA[2].	
T23	P1.05	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
U1	P0.12 TRACEDATA1	Digital I/O Trace data	General purpose I/O. Trace buffer TRACEDATA[1].	
U24	P1.04	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
V23	P1.03	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
W1	VDD	Power	Power supply	
W24	P1.02	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
Y2	VDDH	Power	High voltage power supply	
Y23	P1.01	Digital I/O	General purpose I/O	Standard drive, low frequency I/O only
AA24	SWDCLK	Debug	Serial wire debug clock input for debug and programming	
AB2	DCCH	Power	DC/DC converter output	
AC5	DECUSB	Power	USB 3.3 V regulator supply decoupling	
AC9	P0.14	Digital I/O	General purpose I/O	
AC11	P0.16	Digital I/O	General purpose I/O	
AC13	P0.18 nRESET	Digital I/O	General purpose I/O. Configurable as system RESET.	QSPI/CSN
AC15	P0.19	Digital I/O	General purpose I/O	QSPI/SCK
AC17	P0.21	Digital I/O	General purpose I/O	QSPI
AC19	P0.23	Digital I/O	General purpose I/O	QSPI
AC21	P0.25	Digital I/O	General purpose I/O	
AC24	SWDIO	Debug	Serial wire debug I/O for debug and programming	
AD2	VBUS	Power	5 V input for USB 3.3 V regulator	
AD4	D-	Digital I/O	USB D-	USB
AD6	D+	Digital I/O	USB D+	USB
AD8	P0.13	Digital I/O	General purpose I/O	

Pin	Name	Function	Description	Recommended usage
AD10	P0.15	Digital I/O	General purpose I/O	
AD12	P0.17	Digital I/O	General purpose I/O	
AD14	VDD	Power	Power supply	
AD16	P0.20	Digital I/O	General purpose I/O	
AD18	P0.22	Digital I/O	General purpose I/O	QSPI
AD20	P0.24	Digital I/O	General purpose I/O	
AD22	P1.00	Digital I/O	General purpose I/O.	QSPI
	TRACEDATA0	Trace data	Trace buffer TRACEDATA[0]. Serial wire output (SWO).	
AD23	VDD	Power	Power supply	
Bottom of chip				
Die pad	VSS	Power	Ground pad. Exposed die pad must be connected to ground (VSS) for proper device operation.	

Table 137: aQFN™73 ball assignments

Note: For more information on standard drive, see GPIO — General purpose input/output on page 141. Low frequency I/O is signals with a frequency up to 10 kHz.

7.2 Mechanical specifications

The mechanical specifications for the packages show the dimensions in millimeters.

7.2.1 aQFN73 7 x 7 mm package

Dimensions in millimeters for the aQFN™73 7 x 7 mm package.

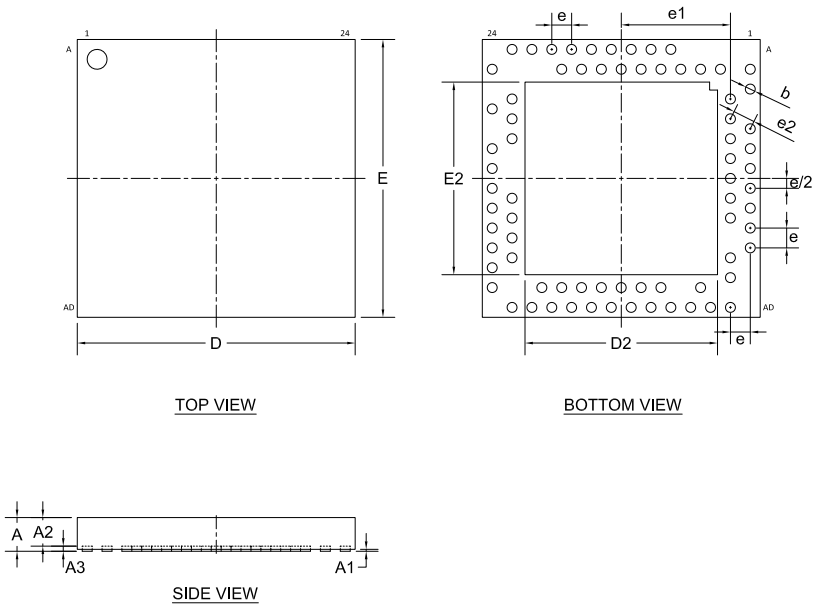

Figure 207: aQFN™73 7 x 7 mm package

	A	A1	A2	A3	b	D, E	D2, E2	e	e1	e2
Min.		0.02			0.20		4.75			
Nom.		0.05	0.675	0.13	0.25	7.00	4.85	0.5	2.75	0.559
Max.	0.85	0.08			0.30		4.95			

Table 138: aQFN™73 dimensions in millimeters

7.3 Reference circuitry

To ensure good RF performance when designing PCBs, it is highly recommended to use the PCB layouts and component values provided by Nordic Semiconductor.

Documentation for the different package reference circuits, including Altium Designer files, PCB layout files, and PCB production files can be downloaded from www.nordicsemi.com.

In this section there are reference circuits for QIAA aQFN™73, showing the components and component values to support on-chip features in a design.

> **Note:** This is not a complete list of configurations, but all required circuitry is shown for further configurations.

Some general guidance is summarized here:

- External supply from VDD is only available when power is supplied to VDDH. External supply is annotated with the VEXT net name.
- When supplying power from a USB source only, VBUS must be connected to VDDH if USB is to be used.
- Components required for DC/DC function are only needed if DC/DC mode is enabled for that regulator.
- NFC can be used in any configuration.
- USB can be used in any configuration as long as VBUS is supplied by the USB host.

Circuit configurations for QIAA aQFN™73

Config no.	Supply configuration		Features that can be enabled for each configuration example				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 1	USB (VDDH = VBUS)	N/A	Yes	No	No	Yes	No
Config. 2	Battery/Ext. regulator	N/A	Yes	No	No	Yes	No
Config. 3	N/A	Battery/Ext. regulator	No	No	No	Yes	No
Config. 4	Battery/Ext. regulator	N/A	Yes	Yes	Yes	Yes	No
Config. 5	N/A	Battery/Ext. regulator	No	No	Yes	Yes	Yes
Config. 6	N/A	Battery/Ext. regulator	No	No	No	No	No

Table 139: Circuit configurations

7.3.1 Circuit configuration no. 1

Circuit configuration number 1 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 1	USB (VDDH = VBUS)	N/A	Yes	No	No	Yes	No

Table 140: Configuration summary for circuit configuration no. 1

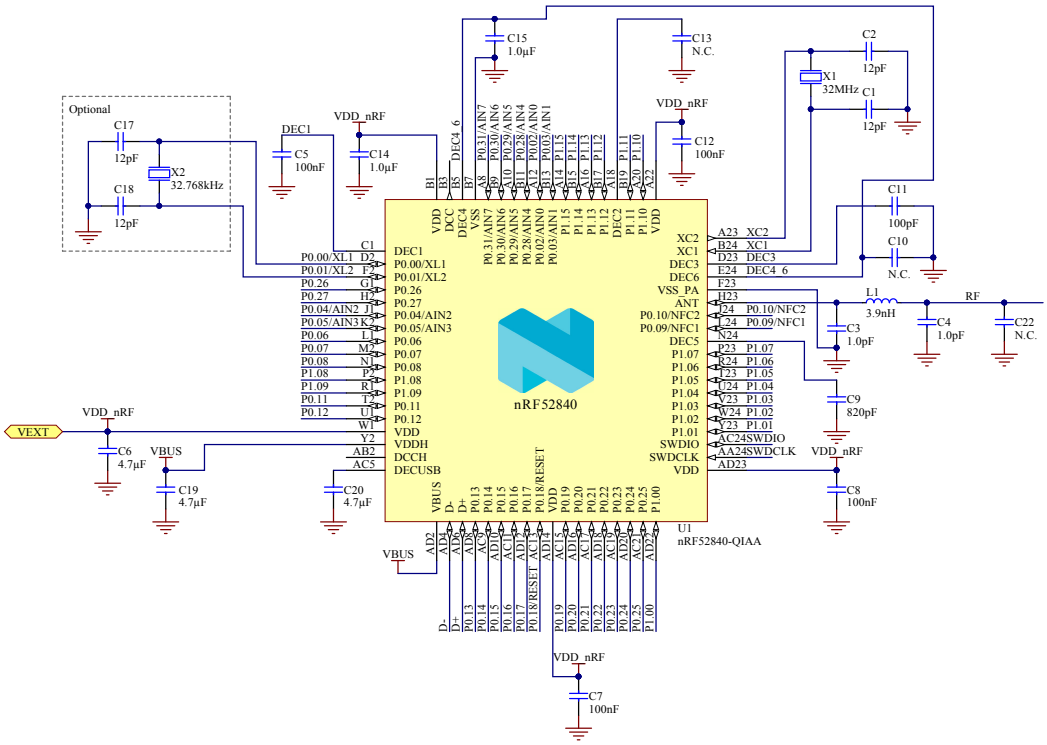

Figure 208: Circuit configuration no. 1 schematic

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2 %	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6, C20	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
C19	4.7 µF	Capacitor, X7S, ± 10 %	0603
L1	3.9 nH	High frequency chip inductor ±5 %	0402
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 141: Bill of material for circuit configuration no. 1

7.3.2 Circuit configuration no. 2

Circuit configuration number 2 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 2	Battery/Ext. regulator	N/A	Yes	No	No	Yes	No

Table 142: Configuration summary for circuit configuration no. 2

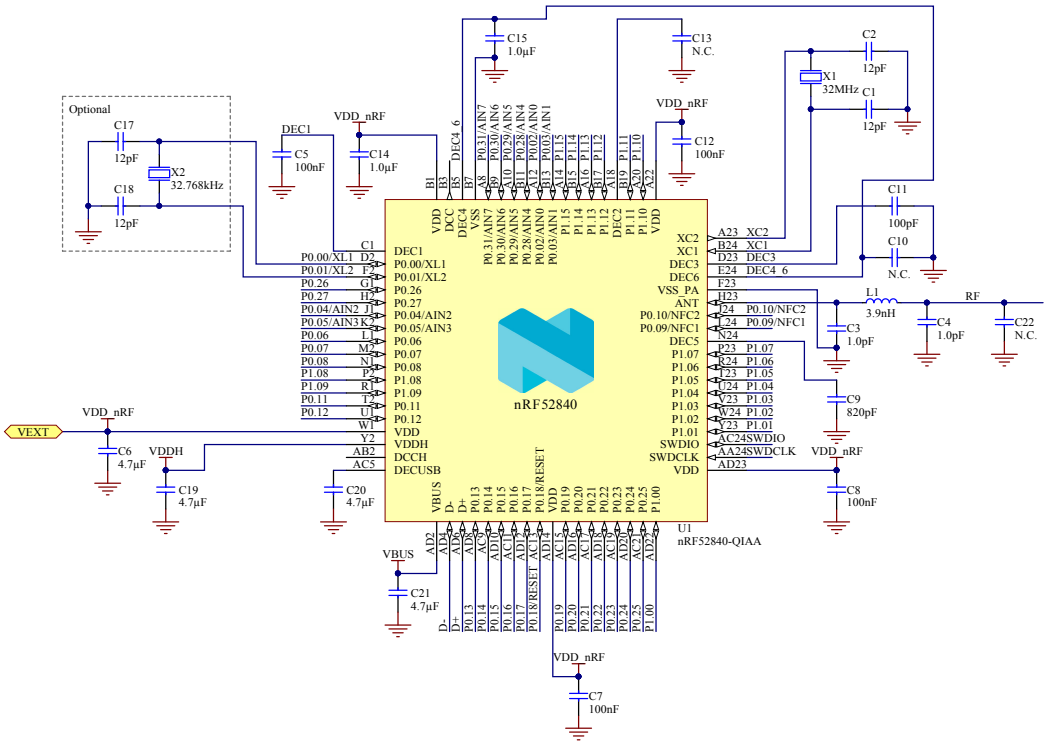

Figure 209: Circuit configuration no. 2 schematic

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2 %	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6, C20	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
C19, C21	4.7 µF	Capacitor, X7S, ±10 %	0603
L1	3.9 nH	High frequency chip inductor ±5 %	0402
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 143: Bill of material for circuit configuration no. 2

7.3.3 Circuit configuration no. 3

Circuit configuration number 3 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 3	N/A	Battery/Ext. regulator	No	No	No	Yes	No

Table 144: Configuration summary for circuit configuration no. 3

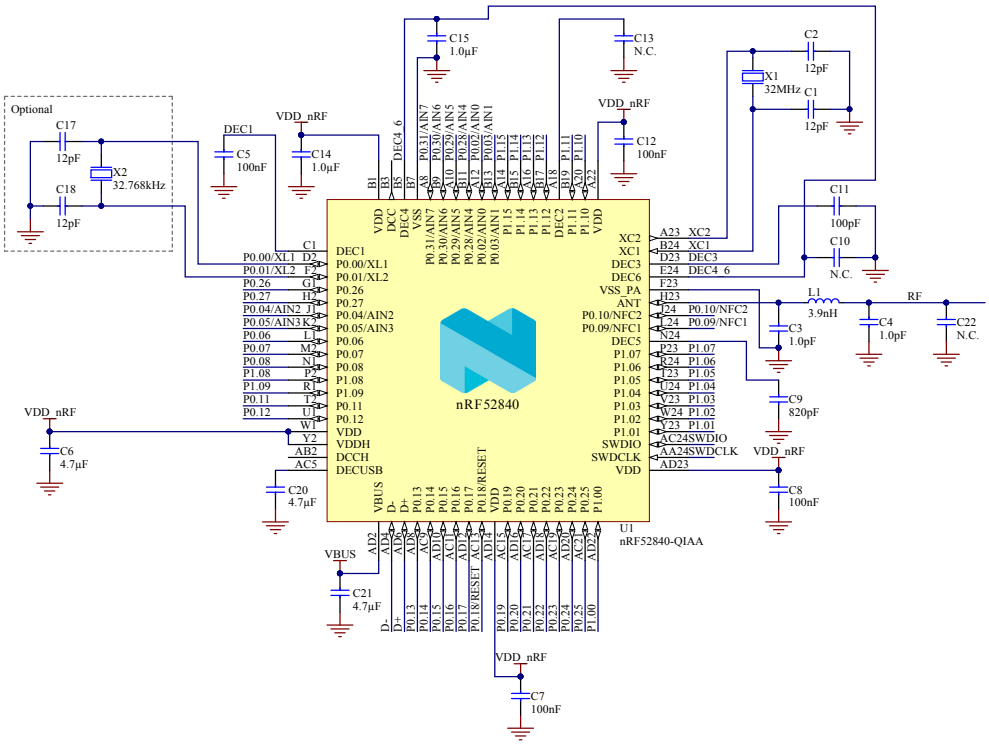

Figure 210: Circuit configuration no. 3 schematic

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2 %	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6, C20	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
C21	4.7 µF	Capacitor, X7S, ±10 %	0603
L1	3.9 nH	High frequency chip inductor ±5 %	0402
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 145: Bill of material for circuit configuration no. 3

7.3.4 Circuit configuration no. 4

Circuit configuration number 4 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 4	Battery/Ext. regulator	N/A	Yes	Yes	Yes	Yes	No

Table 146: Configuration summary for circuit configuration no. 4

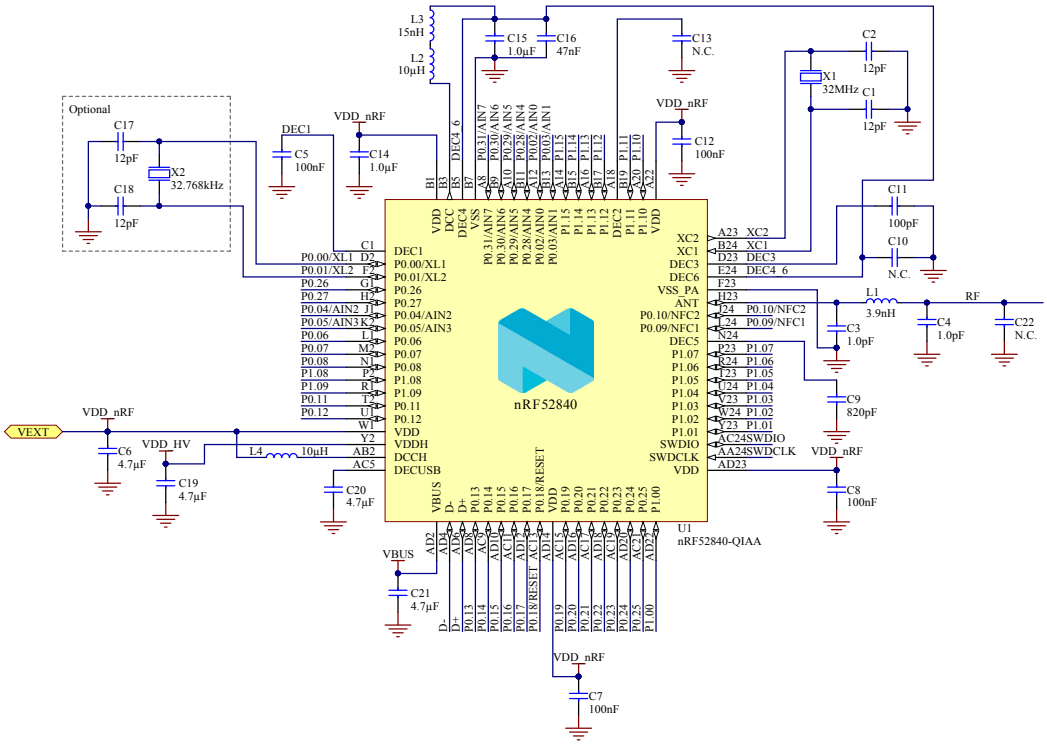

Figure 211: Circuit configuration no. 4 schematic for QIAA aQFN™ 73

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2 %	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6, C20	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
C16	47 nF	Capacitor, X7R, ±10 %	0402
C19, C21	4.7 µF	Capacitor, X7S, ±10 %	0603
L1	3.9 nH	High frequency chip inductor ±5 %	0402
L2	10 µH	Chip inductor, IDC,min = 50 mA, ±20 %	0603
L3	15 nH	High frequency chip inductor ±10 %	0402
L4	10 µH	Chip inductor, IDC,min = 80 mA, ±20 %	0603
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 147: Bill of material for circuit configuration no. 4

7.3.5 Circuit configuration no. 5

Circuit configuration number 5 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 5	N/A	Battery/Ext. regulator	No	No	Yes	Yes	Yes

Table 148: Configuration summary for circuit configuration no. 5

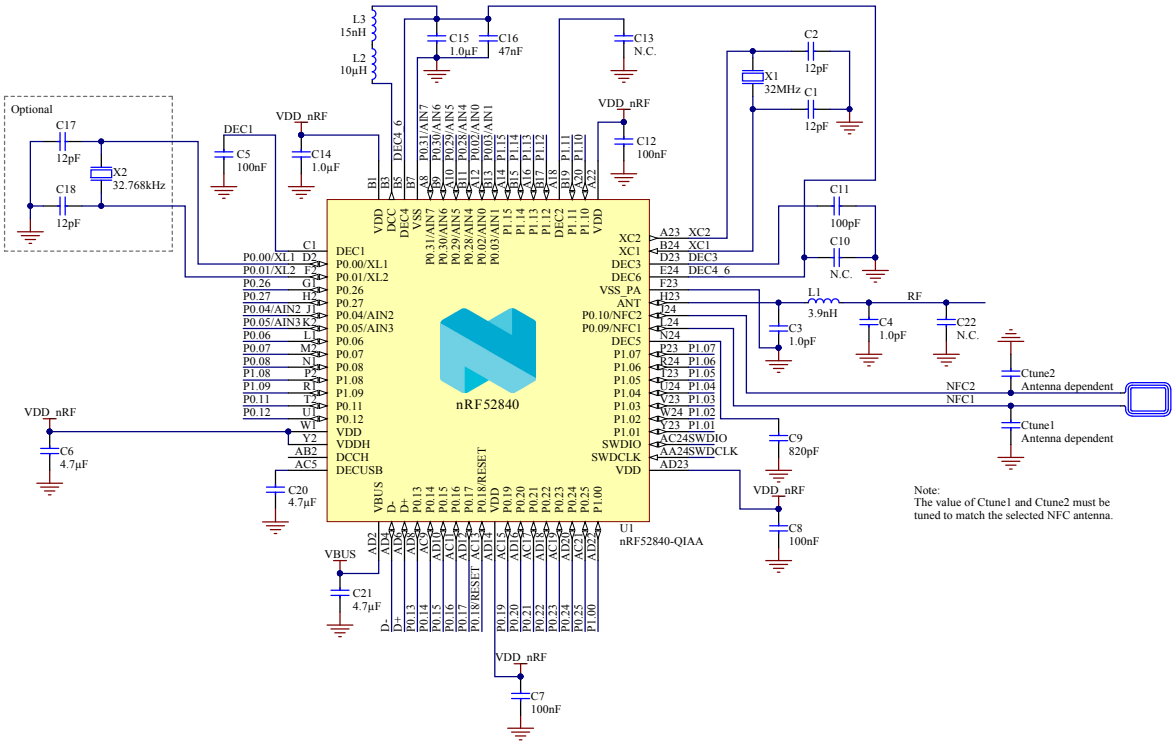

Figure 212: Circuit configuration no. 5 schematic

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2 %	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6, C20	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
C16	47 nF	Capacitor, X7R, ±10 %	0402
C21	4.7 µF	Capacitor, X7S, ±10 %	0603
C_{tune1}, C_{tune2}	Antenna dependent	Capacitor, NP0, ±5 %	0402
L1	3.9 nH	High frequency chip inductor ±5 %	0402
L2	10 µH	Chip inductor, IDC,min = 50 mA, ±20 %	0603
L3	15 nH	High frequency chip inductor ±10 %	0402
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 149: Bill of material for circuit configuration no. 5

7.3.6 Circuit configuration no. 6

Circuit configuration number 6 for QIAA aQFN™73, showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features				
	VDDH	VDD	EXTSUPPLY	DCDCEN0	DCDCEN1	USB	NFC
Config. 6	N/A	Battery/Ext. regulator	No	No	No	No	No

Table 150: Configuration summary for circuit configuration no. 6

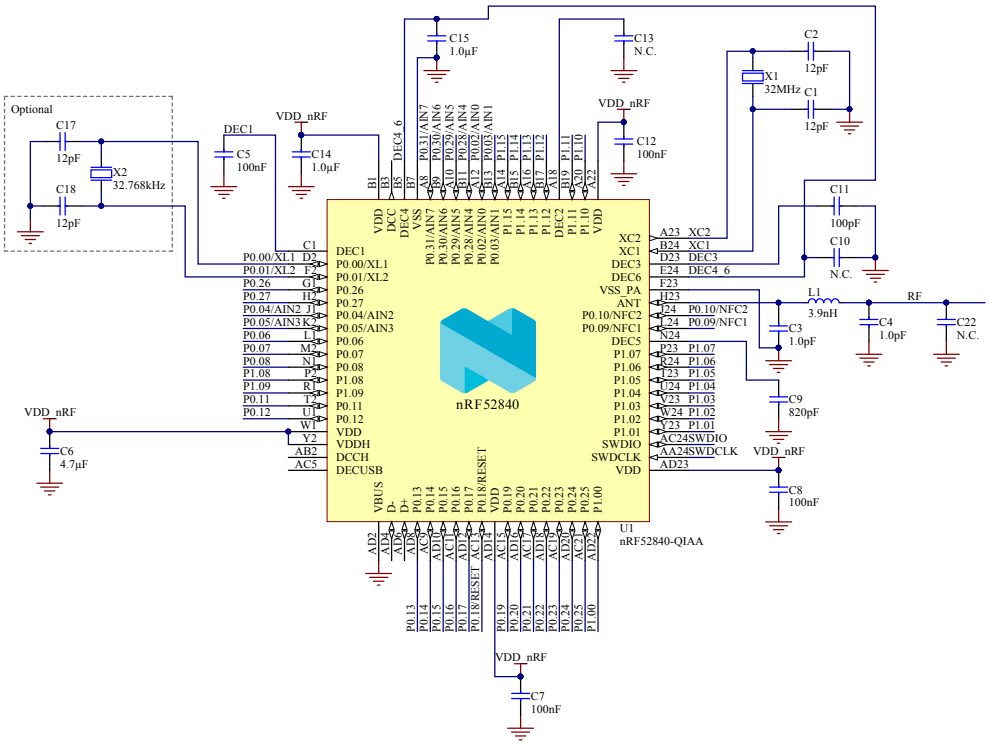

Figure 213: Circuit configuration no. 6 schematic

Note: For PCB reference layouts, see the Reference Layout section on the Downloads tab for the nRF52840 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C17, C18	12 pF	Capacitor, NP0, ±2%	0402
C3, C4	1.0 pF	Capacitor, NP0, ±5 %	0402
C5, C7, C8, C12	100 nF	Capacitor, X7R, ±10 %	0402
C6	4.7 µF	Capacitor, X7R, ±10 %	0603
C9	820 pF	Capacitor, NP0, ±5 %	0402
C10, C13, C22	N.C.	Not mounted	0402
C11	100 pF	Capacitor, NP0, ±5 %	0402
C14, C15	1.0 µF	Capacitor, X7R, ±10 %	0603
L1	3.9 nH	High frequency chip inductor ±5 %	0402
U1	nRF52840-QIAA	Multi-protocol *Bluetooth*® low energy, IEEE 802.15.4, ANT, and 2.4 GHz proprietary system-on-chip	AQFN-73
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_3215

Table 151: Bill of material for circuit configuration no. 6

7.3.7 PCB guidelines

A well designed PCB is necessary to achieve good RF performance. A poor layout can lead to loss in performance or functionality.

A qualified RF layout for the IC and its surrounding components, including matching networks, can be downloaded from www.nordicsemi.com.

To ensure optimal performance it is essential that you follow the schematics- and layout references closely. Especially in the case of the antenna matching circuitry (components between device pin ANT and the antenna), any changes to the layout can change the behavior, resulting in degradation of RF performance or a need to change component values. All the reference circuits are designed for use with a 50 Ω single-ended antenna.

A PCB with a minimum of two layers, including a ground plane, is recommended for optimal RF performance. On PCBs with more than two layers, put a keep-out area on the inner layers directly below the antenna matching circuitry (components between device pin ANT and the antenna) to reduce the stray capacitances that influence RF performance.

A matching network is needed between the RF pin ANT and the antenna, to match the antenna impedance (normally 50 Ω) to the optimum RF load impedance for the chip. For optimum performance, the impedance for the matching network should be set as described in the recommended aQFN™ 73 package reference circuitry from Circuit configuration no. 1 on page 529.

The DC supply voltage should be decoupled as close as possible to the VDD pins with high performance RF capacitors. See the schematics for recommended decoupling capacitor values. The supply voltage for the chip should be filtered and routed separately from the supply voltages of any digital circuitry.

Long power supply lines on the PCB should be avoided. All device grounds, VDD connections, and VDD bypass capacitors must be connected as close as possible to the IC. For a PCB with a topside RF ground

plane, the VSS pins should be connected directly to the ground plane. For a PCB with a bottom ground plane, the best technique is to have via holes as close as possible to the VSS pads. A minimum of one via hole should be used for each VSS pin.

Fast switching digital signals should not be routed close to the crystal or the power supply lines. Capacitive loading of fast switching digital output lines should be minimized in order to avoid radio interference.

7.3.8 PCB layout example

The PCB layout shown below is a reference layout for the aQFN™ package with internal LDO setup and VBUS supply.

> **Note:** Pay attention to how the capacitor C3 is grounded. It is not directly connected to the ground plane, but grounded via VSS_PA pin F23. This is done to create additional filtering of harmonic components.

For all available reference layouts, see the Reference Layout section on the Downloads tab for nRF52840 on www.nordicsemi.com.

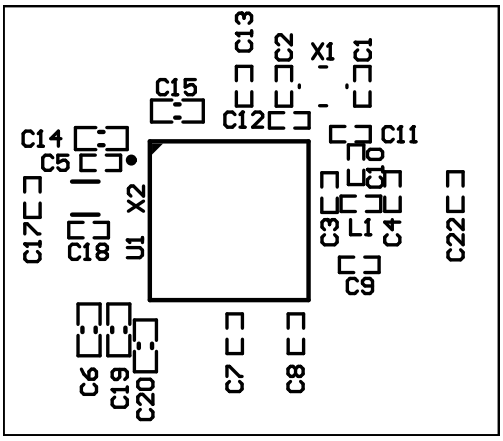

Figure 214: Top silk layer

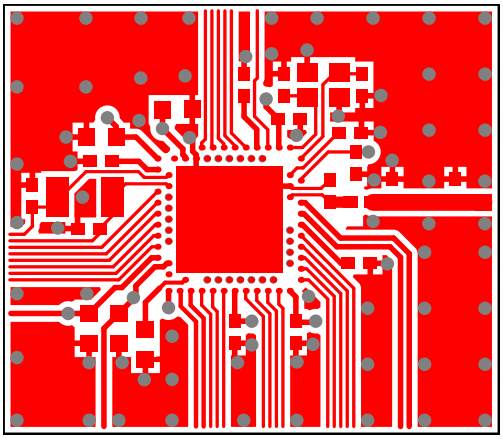

Figure 215: Top layer

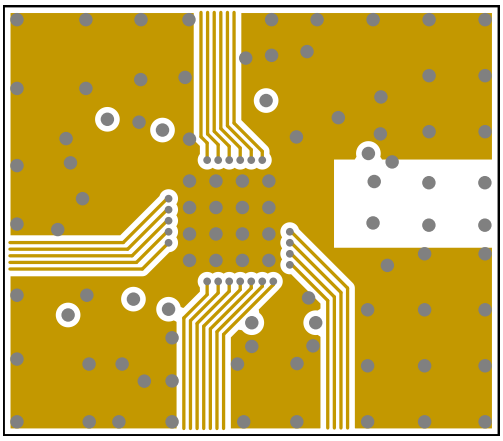

Figure 216: Mid layer 1

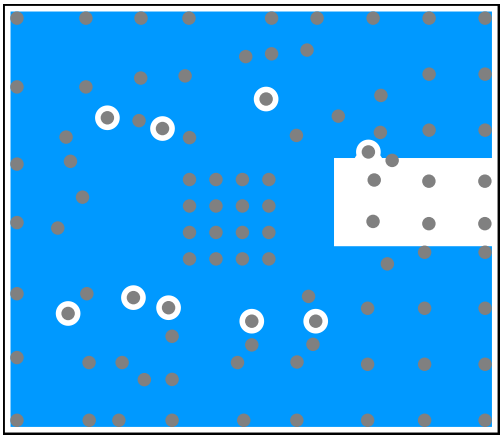

Figure 217: Mid layer 2

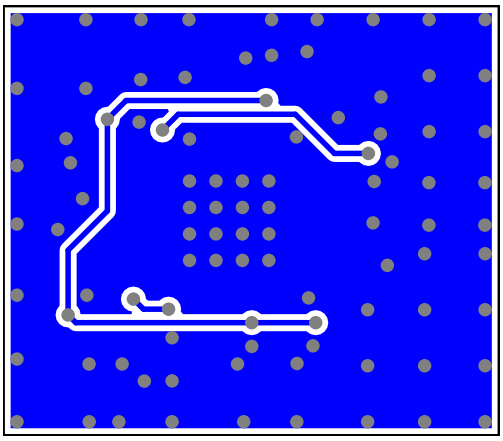

Figure 218: Bottom layer

Note: No components in bottom layer.

8 Recommended operating conditions

The operating conditions are the physical parameters that the chip can operate within.

Symbol	Parameter	Notes	Min.	Nom.	Max.	Units
VDD	VDD supply voltage, independent of DCDC enable		1.7	3.0	3.6	V
VDD_{POR}	VDD supply voltage needed during power-on reset		1.75			V
VDDH	VDDH supply voltage, independent of DCDC enable		2.5	3.7	5.5	V
VBUS	VBUS USB supply voltage		4.35	5	5.5	V
t_{R_VDD}	Supply rise time (0 V to 1.7 V)				60	ms
t_{R_VDDH}	Supply rise time (0 V to 3.7 V)				100	ms
TA	Operating temperature		-40	25	85	°C

Table 152: Recommended operating conditions

Important: The on-chip power-on reset circuitry may not function properly for rise times longer than the specified maximum.

9 Absolute maximum ratings

Maximum ratings are the extreme limits to which the chip can be exposed for a limited amount of time without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect the reliability of the device.

	Note	Min.	Max.	Unit
Supply voltages				
VDD		-0.3	+3.9	V
VDDH		-0.3	+5.8	V
VBUS		-0.3	+5.8	V
VSS			0	V
I/O pin voltage				
$V_{I/O}$, VDD ≤3.6 V		-0.3	VDD + 0.3 V	V
$V_{I/O}$, VDD >3.6 V		-0.3	3.9 V	V
NFC antenna pin current				
$I_{NFC1/2}$			80	mA
Radio				
RF input level			10	dBm
Environmental (aQFN™ package)				
Storage temperature		-40	+125	°C
MSL	Moisture Sensitivity Level		2	
ESD HBM	Human Body Model		2	kV
ESD CDM_{QF}	Charged Device Model (aQFN™ 73, 7×7 mm package)		750	V
Flash memory				
Endurance		10 000		Write/erase cycles
Retention		10 years at 40°C		

Table 153: Absolute maximum ratings

10 Ordering information

This chapter contains information on IC marking, ordering codes, and container sizes.

10.1 Package marking

The nRF52840 SoC package is marked as shown in the figure below.

N	5	2	8	4	0
<P	P>	<V	V>	<H>	<P>
<Y	Y>	<W	W>	<L	L>

Figure 219: Package marking

10.2 Box labels

Here are the box labels used for the nRF52840.

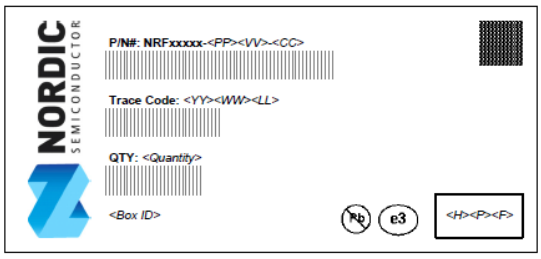

Figure 220: Inner box label

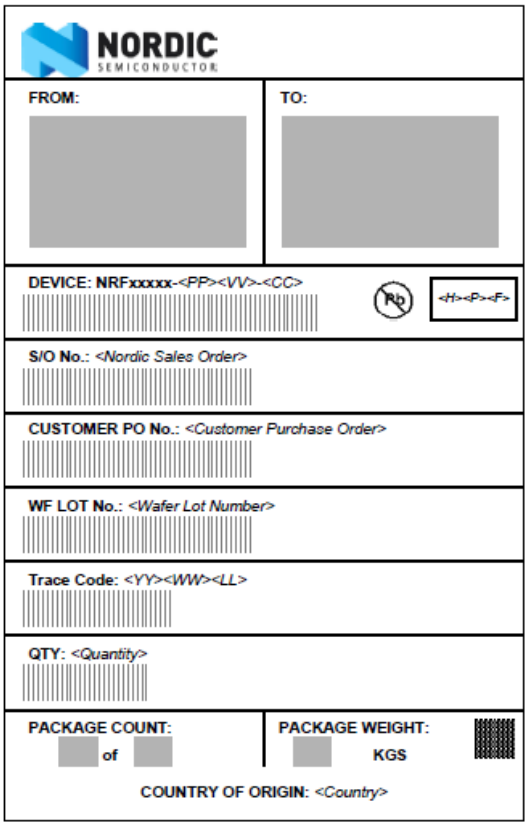

Figure 221: Outer box label

10.3 Order code

Here are the nRF52840 order codes and definitions.

n	R	F	5	2	8	4	0	-	<P	P>	<V	V>	-	<C	C>

Figure 222: Order code

Abbrevitation	Definition and implemented codes
N52/nRF52	nRF52 series product
840	Part code
<PP>	Package variant code
<VV>	Function variant code
<H><P><F>	Build code H - Hardware version code P - Production configuration code (production site, etc.) F - Firmware version code (only visible on shipping container label)
<YY><WW><LL>	Tracking code YY - Year code WW - Assembly week number LL - Wafer lot code
<CC>	Container code

Table 154: Abbreviations

10.4 Code ranges and values

Defined here are the nRF52840 code ranges and values.

<PP>	Package	Size (mm)	Pin/Ball count	Pitch (mm)
QI	aQFN™	7 x 7	73	0.5

Table 155: Package variant codes

<VV>	Flash (kB)	RAM (kB)
AA	1024	256

Table 156: Function variant codes

<H>	Description
[A . . Z]	Hardware version/revision identifier (incremental)

Table 157: Hardware version codes

<P>	Description
[0 . . 9]	Production device identifier (incremental)
[A . . Z]	Engineering device identifier (incremental)

Table 158: Production configuration codes

<F>	Description
[A . . N, P . . Z]	Version of preprogrammed firmware
[0]	Delivered without preprogrammed firmware

Table 159: Production version codes

<YY>	Description
[16 . . 99]	Production year: 2016 to 2099

Table 160: Year codes

<WW>	Description
[1 . . 52]	Week of production

Table 161: Week codes

<LL>	Description
[AA . . ZZ]	Wafer production lot identifier

Table 162: Lot codes

<CC>	Description
R7	7" Reel
R	13" Reel
T	Tray

Table 163: Container codes

10.5 Product options

Defined here are the nRF52840 product options.

Order code	MOQ[43]
nRF52840-QIAA-R7	800
nRF52840-QIAA-R	3000
nRF52840-QIAA-T	260

Table 164: nRF52840 order codes

Order code	Description
nRF52840-DK	nRF52840 Development Kit

Table 165: Development tools order code

[43] Minimum Ordering Quantity

11 Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

11.1 RoHS and REACH statement

Nordic Semiconductor products meet the requirements of Directive 2011/65/EU of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS 2) and the requirements of the REACH regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals.

The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website.

11.2 Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.