nRF52810

Product Specification

v1.0

NORDIC® SEMICONDUCTOR

Key features

Key features:

- 2.4 GHz transceiver
  - -96 dBm sensitivity in *Bluetooth*® low energy mode
  - Supported data rates: 1 Mbps, 2 Mbps *Bluetooth*® low energy mode
  - -20 to +4 dBm TX power, configurable in 4 dB steps
  - On-chip balun (single-ended RF)
  - 4.6 mA peak current in TX (0 dBm)
  - 4.6 mA peak current in RX
  - RSSI (1 dB resolution)
- ARM® Cortex®-M4 32-bit processor, 64 MHz
  - 144 EEMBC CoreMark® score running from flash memory
  - 34.4 µA/MHz running from flash memory
  - 32.8 µA/MHz running from RAM
  - Serial wire debug (SWD)
- Flexible power management
  - 1.7 V-3.6 V supply voltage range
  - Fully automatic LDO and DC/DC regulator system
  - Fast wake-up using 64 MHz internal oscillator
  - 0.3 µA at 3 V in System OFF mode, no RAM retention
  - 0.5 µA at 3 V in System OFF mode with full 24 kB RAM retention
  - 1.5 µA at 3 V in System ON mode, with full 24 kB RAM retention, wake on RTC
- 192 kB flash and 24 kB RAM
- Nordic SoftDevice ready
- Support for concurrent multi-protocol
- 12-bit, 200 ksps ADC - 8 configurable channels with programmable gain
- 64 level comparator
- Temperature sensor
- Up to 32 general purpose I/O pins
- 4-channel pulse width modulator (PWM) unit with EasyDMA
- Digital microphone interface (PDM)
- 3x 32-bit timer with counter mode
- SPI master/slave with EasyDMA
- I2C compatible 2-wire master/slave
- UART (CTS/RTS) with EasyDMA
- Programmable peripheral interconnect (PPI)
- Quadrature decoder (QDEC)
- AES HW encryption with EasyDMA
- 2x real-time counter (RTC)
- Single crystal operation
- Package variants
  - QFN48 package, 6 x 6 mm
  - QFN32 package, 5 x 5 mm

Applications:

- Computer peripherals and I/O devices
  - Mouse
  - Keyboard
  - Mobile HID
- CE remote controls
- Network processor
  - Wearables
  - Virtual reality headsets
- Health and medical
- Enterprise lighting
  - Industrial
  - Commercial
  - Retail
- Beacons
- Connectivity device in multi-chip solutions

Contents

Key features . . . ii

1 Revision history . . . 9

2 About this document . . . 10

2.1 Document naming and status . . . 10
2.2 Peripheral naming and abbreviations . . . 10
2.3 Register tables . . . 11
2.3.1 Fields and values . . . 11
2.4 Registers . . . 11
2.4.1 DUMMY . . . 11

3 Block diagram . . . 13

4 Core components . . . 14

4.1 CPU . . . 14
4.1.1 Electrical specification . . . 14
4.1.2 CPU and support module configuration . . . 14
4.2 Memory . . . 15
4.2.1 RAM - Random access memory . . . 16
4.2.2 Flash - Non-volatile memory . . . 16
4.2.3 Memory map . . . 16
4.2.4 Instantiation . . . 17
4.3 NVMC — Non-volatile memory controller . . . 18
4.3.1 Writing to flash . . . 18
4.3.2 Erasing a page in flash . . . 18
4.3.3 Writing to user information configuration registers (UICR) . . . 18
4.3.4 Erasing user information configuration registers (UICR) . . . 19
4.3.5 Erase all . . . 19
4.3.6 Registers . . . 19
4.3.7 Electrical specification . . . 21
4.4 FICR — Factory information configuration registers . . . 22
4.4.1 Registers . . . 22
4.5 UICR — User information configuration registers . . . 32
4.5.1 Registers . . . 32
4.6 EasyDMA . . . 48
4.6.1 EasyDMA array list . . . 49
4.7 AHB multilayer . . . 50
4.8 Debug . . . 50
4.8.1 DAP - Debug Access Port . . . 51
4.8.2 CTRL-AP - Control Access Port . . . 51
4.8.3 Debug interface mode . . . 53
4.8.4 Real-time debug . . . 54

5 Power and clock management . . . 55

5.1 Power management unit (PMU) . . . 55
5.2 Current consumption . . . 55
5.2.1 Electrical specification . . . 56
5.3 POWER — Power supply . . . 61

7.3 Reference circuitry . . . 468
7.3.1 Schematic QFAA QFN48 with internal LDO setup . . . 468
7.3.2 Schematic QFAA QFN48 with DC/DC regulator setup . . . 470
7.3.3 Schematic QCAA QFN32 with internal LDO setup . . . 471
7.3.4 Schematic QCAA QFN32 with DC/DC regulator setup . . . 472
7.3.5 PCB guidelines . . . 473
7.3.6 PCB layout example . . . 474

8 Recommended operating conditions . . . 476

9 Absolute maximum ratings . . . 477

10 Ordering information . . . 478
10.1 IC marking . . . 478
10.2 Box labels . . . 478
10.3 Order code . . . 479
10.4 Code ranges and values . . . 480
10.5 Product options . . . 481

11 Liability disclaimer . . . 483

1 Revision history

Date	Version	Description
September 2017	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 476.

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**	**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																											
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.

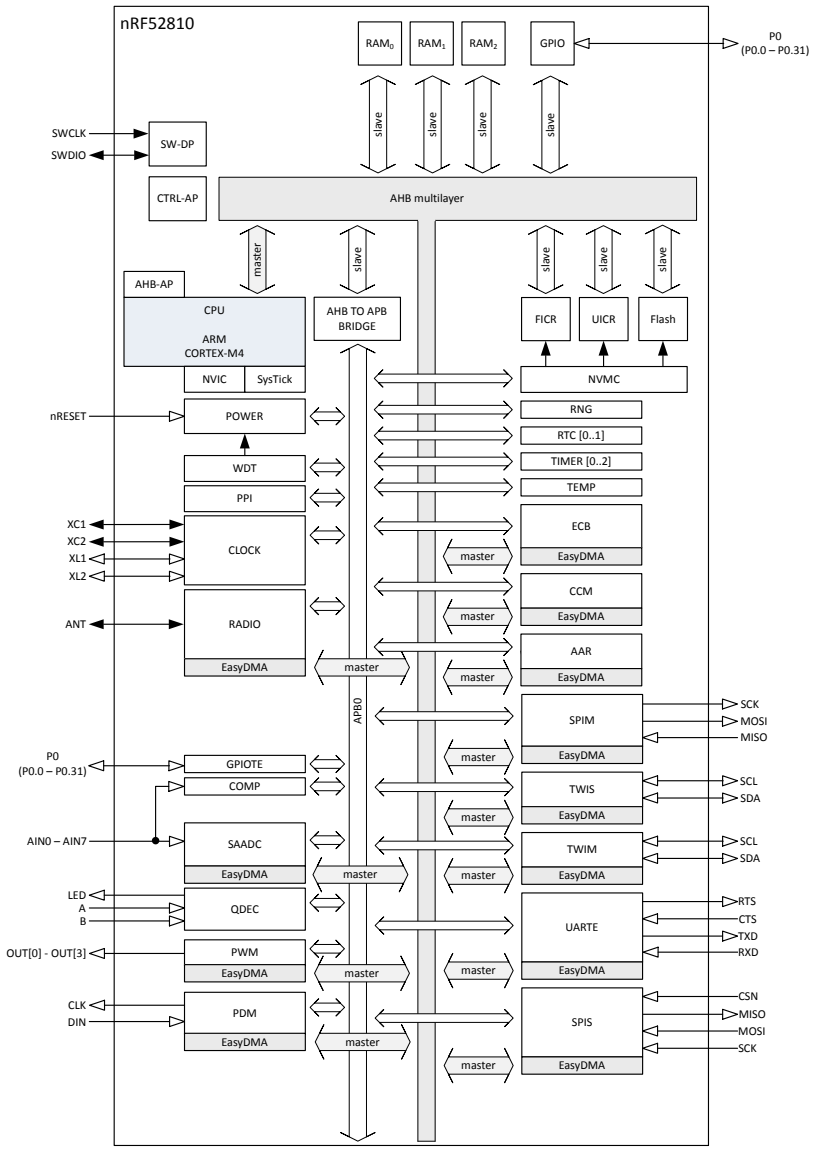

Figure 1: Block diagram

4 Core components

4.1 CPU

The ARM® Cortex®-M4 processor 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

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. The section Electrical specification on page 14 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 the device. The SysTick's clock will only tick when the CPU is running or when the system is in debug interface mode.

4.1.1 Electrical specification

4.1.1.1 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 from flash	0		2	
W_{RAM}	CPU wait states, running from RAM			0	
CM_{FLASH}	CoreMark[1], running from flash		144		CoreMark
$CM_{FLASH/MHz}$	CoreMark per MHz, running from flash		2.25		CoreMark/MHz
$CM_{FLASH/mA}$	CoreMark per mA, running from flash, DCDC 3V		60		CoreMark/mA

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.

[1] Using IAR v6.50.1.4452 with flags --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp –Ohs --no_size_constraints

Option / Module	Description	Implemented
Core options		
NVIC	Nested vector interrupt controller	30 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	NO
SysTick	System tick timer	YES
Modules		
MPU	Memory protection unit	YES
FPU	Floating-point unit	NO
DAP	Debug access port	YES
ETM	Embedded trace macrocell	NO
ITM	Instrumentation trace macrocell	NO
TPIU	Trace port interface unit	NO
ETB	Embedded trace buffer	NO
FPB	Flash patch and breakpoint unit	YES
HTM	AMBA® AHB trace macrocell	NO

4.2 Memory

The nRF52810 contains flash and RAM that can be used for code and data storage.

The amount of RAM and flash will vary depending on variant, see Memory variants on page 15.

Device name	RAM	Flash	Comments
nRF52810-QFAA	24 kB	192 kB	

Table 3: Memory variants

The CPU and the 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 16.

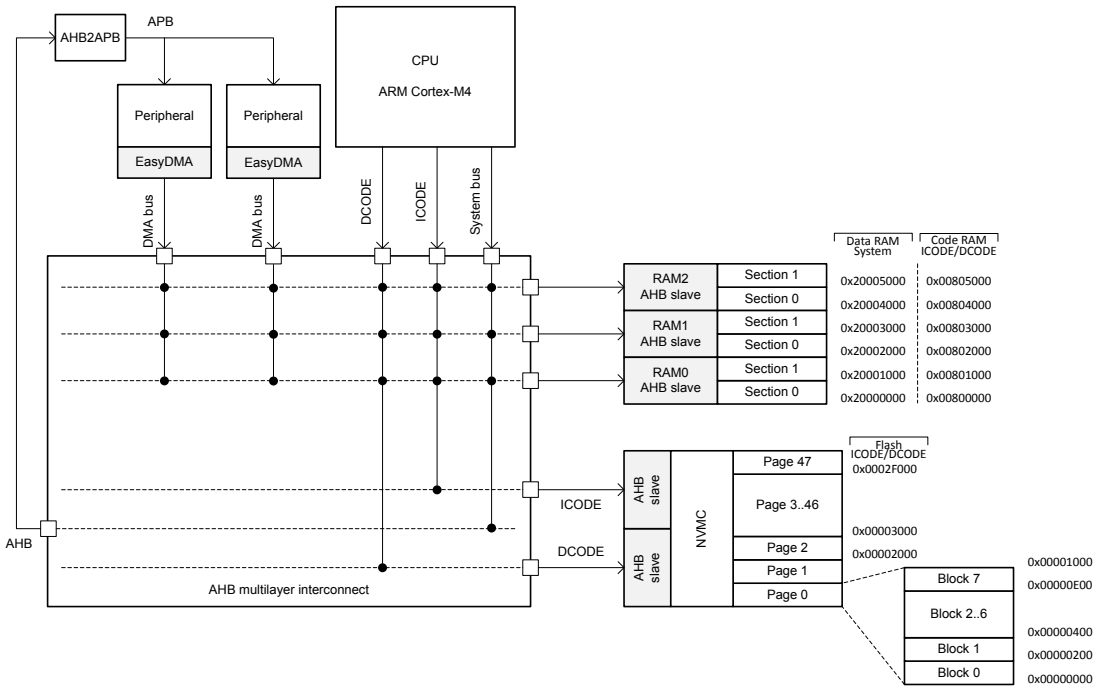

Figure 2: Memory layout

See AHB multilayer on page 50 and EasyDMA on page 48 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 multiple RAM AHB slaves.

Each RAM AHB slave is connected to two 4-kilobyte RAM sections, see Section 0 and Section 1 in Memory layout on page 16.

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 61).

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 18.

The flash is divided into multiple 4 kB pages that can be accessed by the CPU via both the ICODE and DCODE buses as shown in, Memory layout on page 16. Each page is divided into 8 blocks.

4.2.3 Memory map

The complete memory map is shown in Memory map on page 17. As described in Memory on page 15, Code RAM and Data RAM are the same physical RAM.

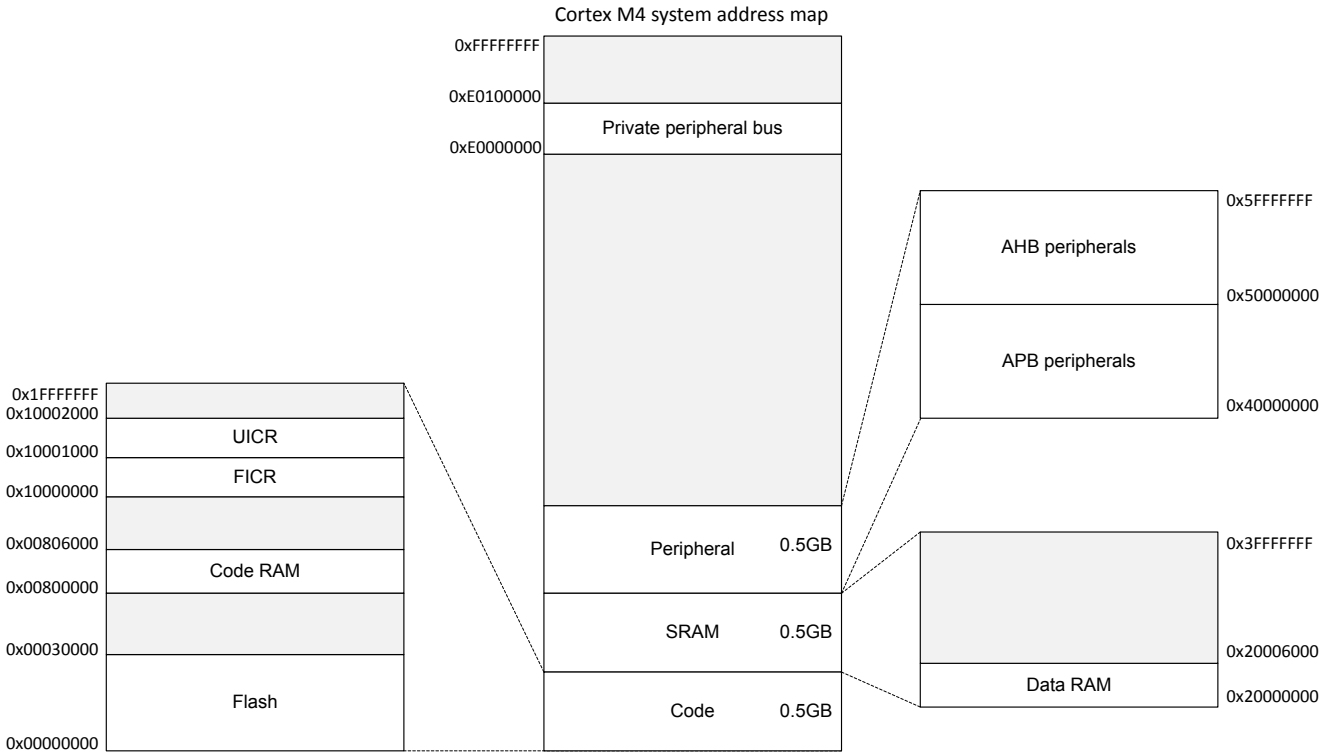

Figure 3: Memory map

4.2.4 Instantiation

ID	Base Address	Peripheral	Instance	Description
0	0x40000000	CLOCK	CLOCK	Clock control
0	0x40000000	BPROT	BPROT	Block protect
0	0x40000000	POWER	POWER	Power control
1	0x40001000	RADIO	RADIO	2.4 GHz radio
2	0x40002000	UARTE	UARTE0	Universal asynchronous receiver/transmitter with EasyDMA
3	0x40003000	TWIM	TWIM0	Two-wire interface master
3	0x40003000	TWIS	TWIS0	Two-wire interface slave
4	0x40004000	SPIS	SPIS0	SPI slave
4	0x40004000	SPIM	SPIM0	SPI master
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 Codebook (ECB) mode block encryption
15	0x4000F000	AAR	AAR	Accelerated address resolver
15	0x4000F000	CCM	CCM	AES CCM mode 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
20	0x40014000	SWI	SWI0	Software interrupt 0
20	0x40014000	EGU	EGU0	Event generator unit 0

ID	Base Address	Peripheral	Instance	Description
21	0x40015000	EGU	EGU1	Event generator unit 1
21	0x40015000	SWI	SWI1	Software interrupt 1
22	0x40016000	SWI	SWI2	Software interrupt 2
23	0x40017000	SWI	SWI3	Software interrupt 3
24	0x40018000	SWI	SWI4	Software interrupt 4
25	0x40019000	SWI	SWI5	Software interrupt 5
28	0x4001C000	PWM	PWM0	Pulse-width modulation unit 0
29	0x4001D000	PDM	PDM	Pulse-density modulation (digital microphone interface)
30	0x4001E000	NVMC	NVMC	Non-volatile memory controller
31	0x4001F000	PPI	PPI	Programmable peripheral interconnect
0	0x50000000	GPIO	P0	General purpose input and output
N/A	0x10000000	FICR	FICR	Factory information configuration
N/A	0x10001000	UICR	UICR	User information configuration

Table 4: 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 register is used to enable the NVMC for writing (CONFIG.WEN) and erasing (CONFIG.EEN), see CONFIG on page 19. 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 writing is enabled, full 32-bit words are written to word-aligned addresses in flash.

As illustrated in Memory on page 15, the flash is divided into multiple pages that in turn are divided into multiple blocks. The same block in flash can only be written n_{WRITE} number of times before an erase must be performed using ERASEPAGE or ERASEALL. See the memory size and organization in Memory on page 15 for block size.

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.

4.3.2 Erasing a page in flash

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

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.

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 take effect after a reset.

UICR can only be written n_{WRITE} number of times before an erase must be performed using ERASEUICR or ERASEALL. 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 register.

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 register. ERASEALL 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 Registers

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

Table 5: Instances

Register	Offset	Description	
READY	0x400	Ready flag	
CONFIG	0x504	Configuration register	
ERASEPCR1	0x508	Register for erasing a page in code area. Equivalent to ERASEPAGE.	Deprecated
ERASEPAGE	0x508	Register for erasing a page in code area	
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	

Table 6: Register Overview

4.3.6.1 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 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	READY			NVMC is ready or busy
			Busy	0	NVMC is busy (ongoing write or erase operation)
			Ready	1	NVMC is ready

4.3.6.2 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 activate erase and write modes only when they are actively used.
			Ren	0	Read only access
			Wen	1	Write enabled
			Een	2	Erase enabled

4.3.6.3 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.

4.3.6.4 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 behavior, e.g. the wrong page may be erased.

4.3.6.5 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 erase of chip

4.3.6.6 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.

4.3.6.7 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

4.3.7 Electrical specification

4.3.7.1 Flash programming

Symbol	Description	Min.	Typ.	Max.	Units
$n_{WRITE,BLOCK}$	Number of writes allowed in a block before erase				
n_{WRITE}	Number of times an address can be written before erase[2]				
$n_{ENDURANCE}$	Write/erase cycles				
t_{WRITE}	Time to write one 32-bit word				µs
$t_{ERASEPAGE}$	Time to erase one page				ms
$t_{ERASEALL}$	Time to erase all flash				ms

[2] The page must be erased when either $n_{WRITE,BLOCK}$ or n_{WRITE} is exceeded.

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	Part variant, hardware version and production configuration	
INFO.PACKAGE	0x108	Package option	
INFO.RAM	0x10C	RAM variant	
INFO.FLASH	0x110	Flash variant	
	0x114		Reserved
	0x118		Reserved
	0x11C		Reserved
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	
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	

Register	Offset	Description
TEMP.T4	0x444	Segment end T4

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
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**

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
Reset 0x00000030	**0 0 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 0 0**

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[0]

Address offset: 0x060

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
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 DEVICEID[1]

Address offset: 0x064

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.

4.4.1.5 ER[0]

Address offset: 0x080

Encryption root, word 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 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

4.4.1.6 ER[1]

Address offset: 0x084

Encryption root, word 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	ER			Encryption root, word n

4.4.1.7 ER[2]

Address offset: 0x088

Encryption root, word 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	ER			Encryption root, word n

4.4.1.8 ER[3]

Address offset: 0x08C

Encryption root, word 3

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field	Value Id	Value	Description																											
A	R	ER			Encryption root, word n																											

4.4.1.9 IR[0]

Address offset: 0x090

Identity root, word 0

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field	Value Id	Value	Description																											
A	R	IR			Identity root, word n																											

4.4.1.10 IR[1]

Address offset: 0x094

Identity root, word 1

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field	Value Id	Value	Description																											
A	R	IR			Identity root, word n																											

4.4.1.11 IR[2]

Address offset: 0x098

Identity root, word 2

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field	Value Id	Value	Description																											
A	R	IR			Identity root, word n																											

4.4.1.12 IR[3]

Address offset: 0x09C

Identity root, word 3

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field	Value Id	Value	Description																											
A	R	IR			Identity root, word n																											

4.4.1.13 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

4.4.1.14 DEVICEADDR[0]

Address offset: 0x0A4

Device 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 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.

4.4.1.15 DEVICEADDR[1]

Address offset: 0x0A8

Device 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 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.

4.4.1.16 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 0x00052810				**0 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 0 0 1 0 0 0 0**
Id	RW	Field	Value Id	Value	Description
A	R	PART			Part code
			N52810	0x52810	nRF52810
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.17 INFO.VARIANT

Address offset: 0x104

Part variant, 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			Part variant, hardware version and production configuration, encoded as ASCII
			AAAA	0x41414141	AAAA
			AAA0	0x41414130	AAA0
			AABA	0x41414241	AABA
			AABB	0x41414242	AABB
			AAB0	0x41414230	AAB0
			AACA	0x41414341	AACA
			AACB	0x41414342	AACB
			AAC0	0x41414330	AAC0
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.18 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
			QF	0x2000	QFxx - 48-pin QFN
			QC	0x2003	QCxx - 32-pin QFN
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.19 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
Reset 0x00000018				**0 0 0 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 0**
Id	RW	Field	Value Id	Value	Description
A	R	RAM			RAM variant
			K24	0x18	24 kByte RAM
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.20 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
Reset 0x000000C0				**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 0 0 0 0**
Id	RW	Field	Value Id	Value	Description
A	R	FLASH			Flash variant
			K192	0xC0	192 kByte flash
			Unspecified	0xFFFFFFFF	Unspecified

4.4.1.21 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
Reset 0x00000320				**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 0 0 0**
Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.22 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
Reset 0x00000343				**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 0 0 1 1**
Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.23 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
Reset 0x0000035D	**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 1 1 1 0 1**

Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.24 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
Reset 0x00000400	**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 0**

Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.25 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
Reset 0x00000452	**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 1 0 1 0 0 1 0**

Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.26 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
Reset 0x0000037B	**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 1 1 1 0 1 1**

Id	RW	Field	Value Id	Value	Description
A	R	A			A (slope definition) register

4.4.1.27 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
Reset 0x00003FCC	**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 0 1 1 0 0**

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

4.4.1.28 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 0x00003F98	**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 0 1 1 0 0 0**

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

4.4.1.29 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 A A
Reset 0x00003F98	**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 0 1 1 0 0 0**

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

4.4.1.30 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 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	R	B			B (y-intercept)

4.4.1.31 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 A A
Reset 0x0000004D	**0 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 1 0 1**

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

4.4.1.32 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 0x00003E10	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0**

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

4.4.1.33 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 0x000000E2	**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	R	T			T (segment end) register

4.4.1.34 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 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	T			T (segment end) register

4.4.1.35 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 0x00000014	**0 0 0 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**

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

4.4.1.36 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 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	R	T			T (segment end) register

4.4.1.37 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 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	R	T			T (segment end) register

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 18 and Memory on page 15 chapters.

4.5.1 Registers

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

Table 9: Instances

Register	Offset	Description	
	0x000		Reserved
	0x004		Reserved
	0x008		Reserved
	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	

Register	Offset	Description
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
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 (see POWER chapter for details)
PSELRESET[1]	0x204	Mapping of the nRESET function (see POWER chapter for details)
APPROTECT	0x208	Access port protection

Table 10: Register Overview

4.5.1.1 NRFFW[0]

Address offset: 0x014

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 NRFFW[1]

Address offset: 0x018

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

4.5.1.3 NRFFW[2]

Address offset: 0x01C

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

4.5.1.4 NRFFW[3]

Address offset: 0x020

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

4.5.1.5 NRFFW[4]

Address offset: 0x024

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

4.5.1.6 NRFFW[5]

Address offset: 0x028

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

4.5.1.7 NRFFW[6]

Address offset: 0x02C

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

4.5.1.8 NRFFW[7]

Address offset: 0x030

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

4.5.1.9 NRFFW[8]

Address offset: 0x034

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

4.5.1.10 NRFFW[9]

Address offset: 0x038

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

4.5.1.11 NRFFW[10]

Address offset: 0x03C

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

4.5.1.12 NRFFW[11]

Address offset: 0x040

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

4.5.1.13 NRFFW[12]

Address offset: 0x044

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

4.5.1.14 NRFFW[13]

Address offset: 0x048

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

4.5.1.15 NRFFW[14]

Address offset: 0x04C

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

4.5.1.16 NRFHW[0]

Address offset: 0x050

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

4.5.1.17 NRFHW[1]

Address offset: 0x054

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

4.5.1.18 NRFHW[2]

Address offset: 0x058

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

4.5.1.19 NRFHW[3]

Address offset: 0x05C

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

4.5.1.20 NRFHW[4]

Address offset: 0x060

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

4.5.1.21 NRFHW[5]

Address offset: 0x064

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

4.5.1.22 NRFHW[6]

Address offset: 0x068

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

4.5.1.23 NRFHW[7]

Address offset: 0x06C

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

4.5.1.24 NRFHW[8]

Address offset: 0x070

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

4.5.1.25 NRFHW[9]

Address offset: 0x074

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

4.5.1.26 NRFHW[10]

Address offset: 0x078

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

4.5.1.27 NRFHW[11]

Address offset: 0x07C

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

4.5.1.28 CUSTOMER[0]

Address offset: 0x080

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

4.5.1.29 CUSTOMER[1]

Address offset: 0x084

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

4.5.1.30 CUSTOMER[2]

Address offset: 0x088

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

4.5.1.31 CUSTOMER[3]

Address offset: 0x08C

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

4.5.1.32 CUSTOMER[4]

Address offset: 0x090

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

4.5.1.33 CUSTOMER[5]

Address offset: 0x094

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

4.5.1.34 CUSTOMER[6]

Address offset: 0x098

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

4.5.1.35 CUSTOMER[7]

Address offset: 0x09C

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

4.5.1.36 CUSTOMER[8]

Address offset: 0x0A0

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

4.5.1.37 CUSTOMER[9]

Address offset: 0x0A4

Reserved for customer

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field		Value Id																												
A	RW	CUSTOMER																														

Reserved for customer

4.5.1.38 CUSTOMER[10]

Address offset: 0x0A8

Reserved for customer

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field		Value Id																												
A	RW	CUSTOMER																														

Reserved for customer

4.5.1.39 CUSTOMER[11]

Address offset: 0x0AC

Reserved for customer

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field		Value Id																												
A	RW	CUSTOMER																														

Reserved for customer

4.5.1.40 CUSTOMER[12]

Address offset: 0x0B0

Reserved for customer

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field		Value Id																												
A	RW	CUSTOMER																														

Reserved for customer

4.5.1.41 CUSTOMER[13]

Address offset: 0x0B4

Reserved for customer

Bit number																																
Id																																
Reset 0xFFFFFFFF																																
Id	RW	Field		Value Id																												
A	RW	CUSTOMER																														

Reserved for customer

4.5.1.42 CUSTOMER[14]

Address offset: 0x0B8

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

4.5.1.43 CUSTOMER[15]

Address offset: 0x0BC

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

4.5.1.44 CUSTOMER[16]

Address offset: 0x0C0

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

4.5.1.45 CUSTOMER[17]

Address offset: 0x0C4

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

4.5.1.46 CUSTOMER[18]

Address offset: 0x0C8

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

4.5.1.47 CUSTOMER[19]

Address offset: 0x0CC

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

4.5.1.48 CUSTOMER[20]

Address offset: 0x0D0

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

4.5.1.49 CUSTOMER[21]

Address offset: 0x0D4

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

4.5.1.50 CUSTOMER[22]

Address offset: 0x0D8

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

4.5.1.51 CUSTOMER[23]

Address offset: 0x0DC

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

4.5.1.52 CUSTOMER[24]

Address offset: 0x0E0

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

4.5.1.53 CUSTOMER[25]

Address offset: 0x0E4

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

4.5.1.54 CUSTOMER[26]

Address offset: 0x0E8

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

4.5.1.55 CUSTOMER[27]

Address offset: 0x0EC

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

4.5.1.56 CUSTOMER[28]

Address offset: 0x0F0

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

4.5.1.57 CUSTOMER[29]

Address offset: 0x0F4

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

4.5.1.58 CUSTOMER[30]

Address offset: 0x0F8

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

4.5.1.59 CUSTOMER[31]

Address offset: 0x0FC

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

4.5.1.60 PSELRESET[0]

Address offset: 0x200

Mapping of the nRESET function (see POWER chapter for details)

All PSELRESET registers have to contain the same value for a pin mapping to be valid. If values are not the same, there will be no nRESET function exposed on a GPIO. As a result, 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				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**
Id	RW	Field	Value Id	Value	Description
A	RW	PIN		21	GPIO number P0.n onto which reset is exposed
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

4.5.1.61 PSELRESET[1]

Address offset: 0x204

Mapping of the nRESET function (see POWER chapter for details)

All PSELRESET registers have to contain the same value for a pin mapping to be valid. If values are not the same, there will be no nRESET function exposed on a GPIO. As a result, 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				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**
Id	RW	Field	Value Id	Value	Description
A	RW	PIN		21	GPIO number P0.n onto which reset is exposed
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

4.5.1.62 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**
Id	RW	Field	Value Id	Value	Description
A	RW	PALL			Enable or disable access port protection. See Debug on page 50 for more information.
			Disabled	0xFF	Disable
			Enabled	0x00	Enable

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 48.

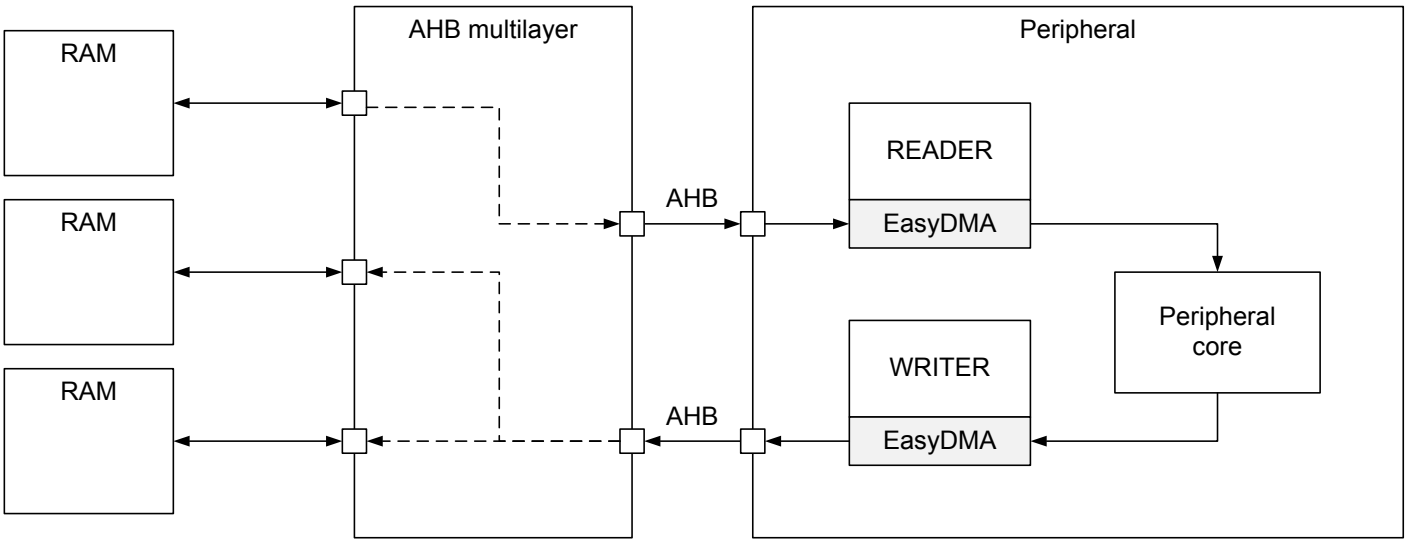

Figure 4: EasyDMA example

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 49.

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.

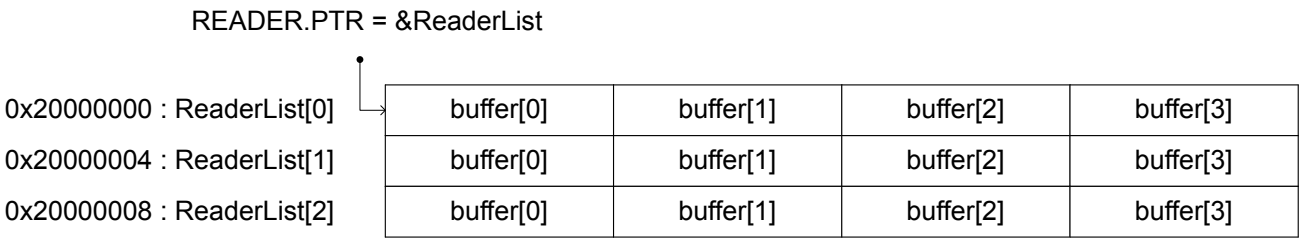

Figure 6: EasyDMA array list

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.

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

Bus master name	Description
CPU	
SPIM0/SPIS0	Same priority and mutually exclusive
RADIO	
CCM/ECB/AAR	Same priority and mutually exclusive
SAADC	
UARTE0	
TWIM0/TWIS0	Same priority and mutually exclusive
PDM	
PWM	

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 15.

4.8 Debug

The debug system offers a flexible and powerful mechanism for non-intrusive debugging.

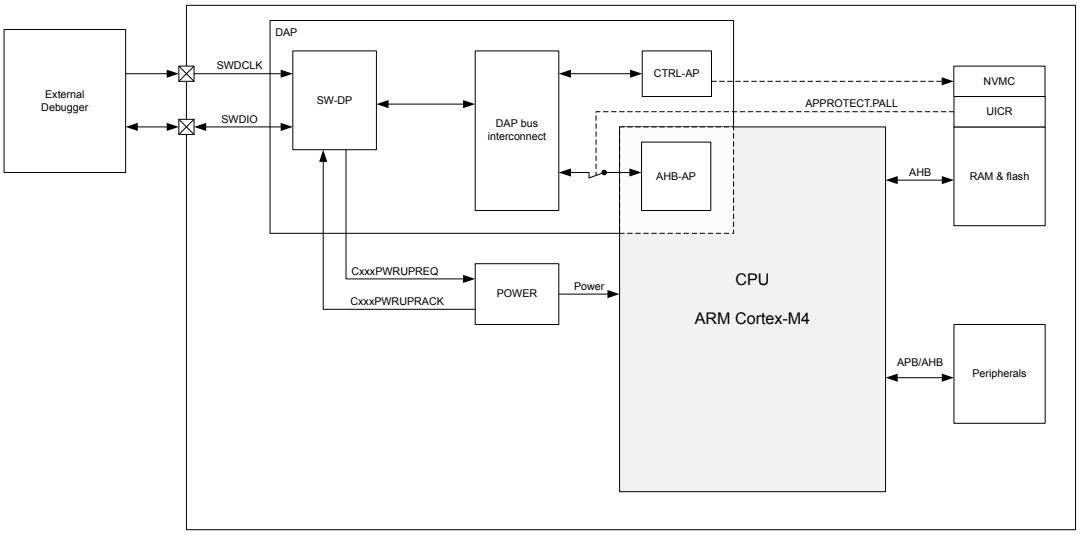

Figure 7: Overview

The main features of the debug system are:

- Two-pin Serial Wire Debug (SWD) interface
- Flash Patch and Breakpoint Unit (FPB) supports:
  - Two literal comparators
  - Six instruction comparators

4.8.1 DAP - Debug Access Port

An external debugger can access the device via the DAP.

The DAP implements a standard ARM® CoreSight™ Serial Wire Debug Port (SW-DP).

The SW-DP implements the Serial Wire Debug protocol (SWD) that is a two-pin serial interface, see SWDCLK and SWDIO in Overview on page 51.

In addition to the default access port in the 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 51.

Important:

- 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 even if the other access ports in the DAP are being 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 47 for more information about enabling access port protection.

This access port enables the following features:

- Soft reset, see Reset on page 65 for more information
- Disable access port protection

Access port protection can only be 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 Behaviour 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 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 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 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 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

4.8.3 Debug interface mode

Before the external debugger can access the 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.

As long as the debugger is requesting power via CxxxPWRUPREQ, the device will be in debug interface mode. If the debugger is not requesting power via CxxxPWRUPREQ, the device will be in normal mode.

Some peripherals will 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 will be higher in debug interface mode compared to normal mode.

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

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 68 will be set.

4.8.4 Real-time debug

The nRF52810 supports real-time debugging.

Real-time debugging will allow interrupts to execute to completion in real time when breakpoints are set in Thread mode or lower priority interrupts. This enables the developer to set a breakpoint and single-step through their code without a failure of the real-time event-driven threads running at higher priority. 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.

5 Power and clock management

5.1 Power management unit (PMU)

Power and clock management in nRF52810 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 55.

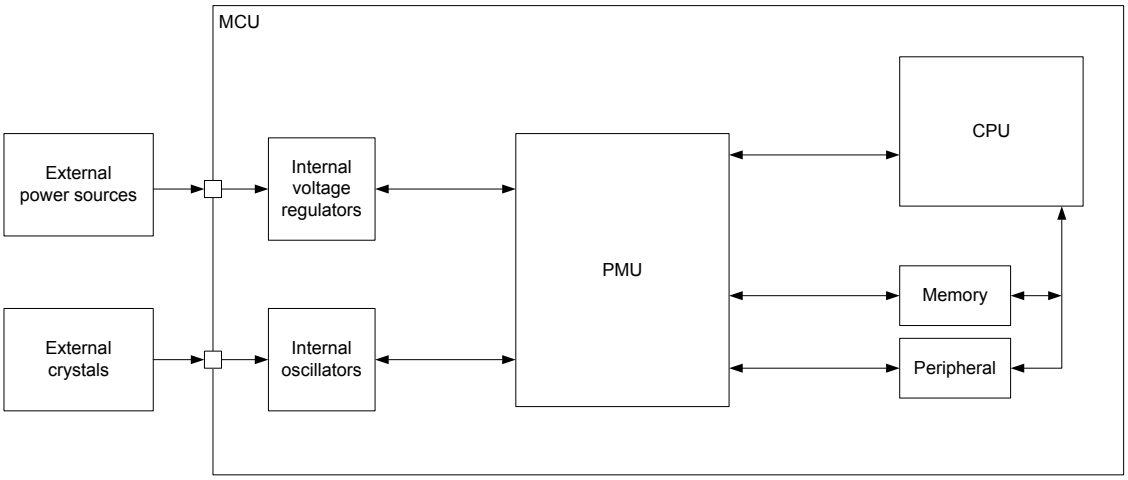

Figure 8: Power management unit

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 55, 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 56 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 56

Condition	Value
VDD	3 V
Temperature	25°C
CPU	WFI (wait for interrupt)/WFE (wait for event) sleep
Peripherals	All idle
Clock	Not running
Regulator	LDO
RAM	Full 24 kB retention
Compiler[3]	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.
32 MHz crystal[4]	SMD 2520, 32 MHz, 10 pF +/- 10 ppm

Table 13: Current consumption scenarios, common conditions

5.2.1 Electrical specification

5.2.1.1 CPU running

Symbol	Description	Min.	Typ.	Max.	Units
I_{CPU0}	CPU running CoreMark @64 MHz from flash, Clock = HFXO, Regulator = DCDC		2.2		mA
I_{CPU1}	CPU running CoreMark @64 MHz from flash, Clock = HFXO		4.2		mA
I_{CPU2}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO, Regulator = DCDC		2.1		mA
I_{CPU3}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO		4		mA
I_{CPU4}	CPU running CoreMark @64 MHz from flash, Clock = HFINT, Regulator = DCDC		2		mA

[3] Applying only when CPU is running
[4] Applying only when HFXO is running

5.2.1.2 Radio transmitting/receiving

Symbol	Description	Min.	Typ.	Max.	Units
I_{RADIO_TX0}	Radio transmitting @ 4 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC		8		mA
I_{RADIO_TX1}	Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC		5.8		mA
I_{RADIO_TX2}	Radio transmitting @ -40 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC		3.4		mA
I_{RADIO_RX0}	Radio receiving @ 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC		6.1		mA
I_{RADIO_TX3}	Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO		10.5		mA
I_{RADIO_TX4}	Radio transmitting @ -40 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO		5.1		mA
I_{RADIO_RX1}	Radio receiving @ 1 Mbps Bluetooth low energy mode, Clock = HFXO		10.8		mA

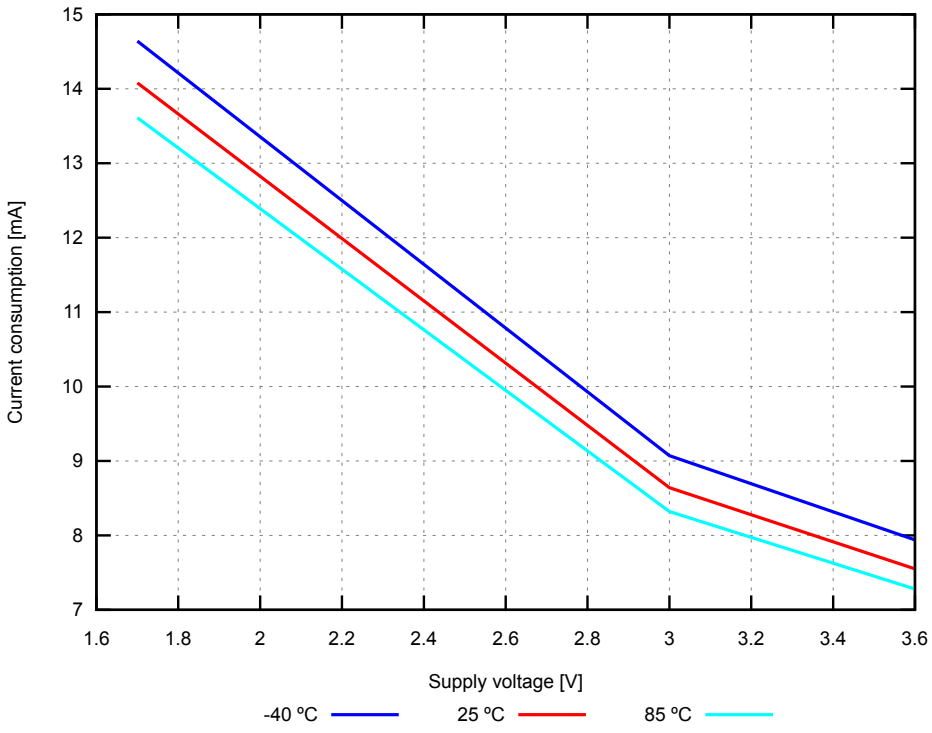

Figure 9: Radio transmitting @ 4 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC (typical values)

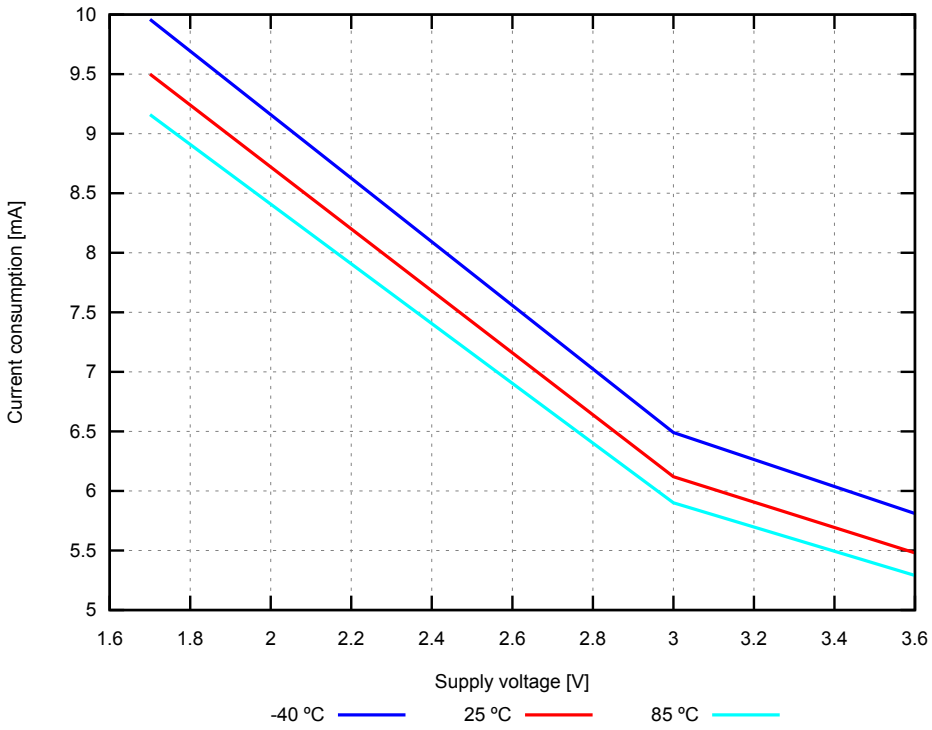

Figure 10: Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC (typical values)

5.2.1.3 Sleep

Symbol	Description	Min.	Typ.	Max.	Units
$I_{ON\_RAMOFF\_EVENT}$	System ON, No RAM retention, Wake on any event		0.6		µA
$I_{ON\_RAMON\_EVENT}$	System ON, Full 24 kB RAM retention, Wake on any event		0.8		µA
$I_{ON\_RAMON\_POF}$	System ON, Full 24 kB RAM retention, Wake on any event, Power fail comparator enabled		0.8		µA
$I_{ON\_RAMON\_GPIOTE}$	System ON, Full 24 kB RAM retention, Wake on GPIOTE input (Event mode)		3.3		µA
$I_{ON\_RAMON\_GPIOTEPORT}$	System ON, Full 24 kB RAM retention, Wake on GPIOTE PORT event		0.8		µA
$I_{ON\_RAMON\_RTC}$	System ON, Full 24 kB RAM retention, Wake on RTC (running from LFRC clock)		1.5		µA
$I_{OFF\_RAMOFF\_RESET}$	System OFF, No RAM retention, Wake on reset		0.3		µA
$I_{OFF\_RAMON\_RESET}$	System OFF, Full 24 kB RAM retention, Wake on reset		0.5		µA

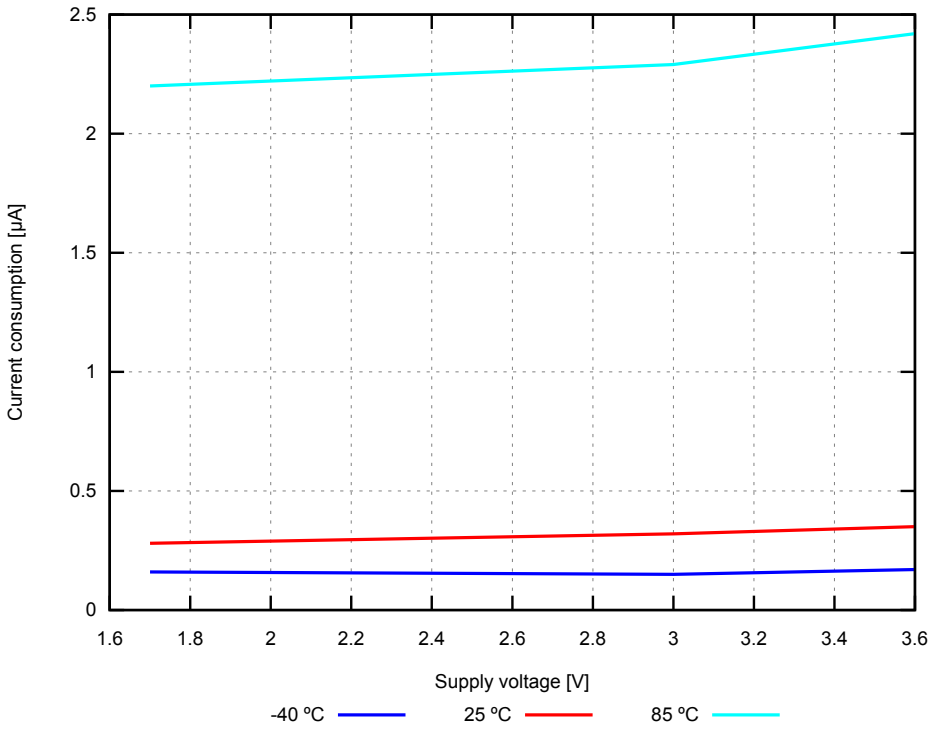

Figure 11: System OFF, No RAM retention, Wake on reset (typical values)

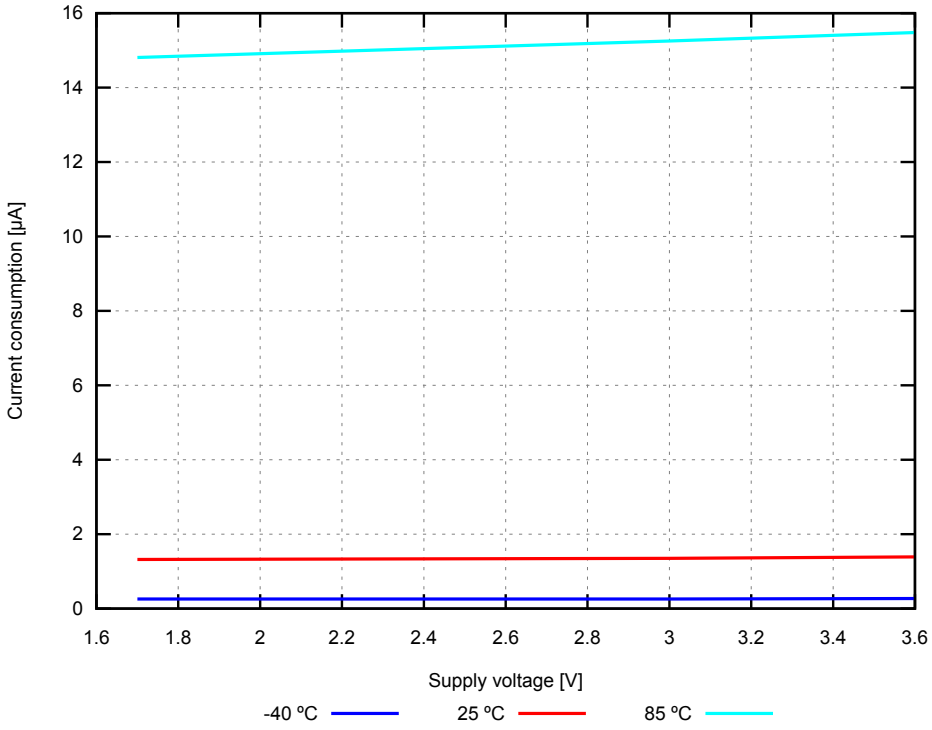

Figure 12: System ON, Full 24 kB RAM retention, Wake on any event (typical values)

5.2.1.4 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 mode, Clock = HFXO, Regulator = DCDC		7.4		mA
I_{S1}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps Bluetooth low energy mode, Clock = HFXO, Regulator = DCDC		7.6		mA
I_{S2}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy mode, Clock = HFXO		13.8		mA
I_{S3}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps Bluetooth low energy mode, Clock = HFXO		14.2		mA

5.2.1.5 TIMER running

Symbol	Description	Min.	Typ.	Max.	Units
I_{TIMER0}	One TIMER instance running @ 1 MHz, Clock = HFINT		432		µA
I_{TIMER1}	Two TIMER instances running @ 1 MHz, Clock = HFINT		432		µA
I_{TIMER2}	One TIMER instance running @ 1 MHz, Clock = HFXO		730		µA
I_{TIMER3}	One TIMER instance running @ 16 MHz, Clock = HFINT		495		µA
I_{TIMER4}	One TIMER instance running @ 16 MHz, Clock = HFXO		792		µA

5.2.1.6 RNG active

Symbol	Description	Min.	Typ.	Max.	Units
I_{RNG0}	RNG running		539		µA

5.2.1.7 TEMP active

Symbol	Description	Min.	Typ.	Max.	Units
I_{TEMP0}	TEMP started		998		µA

5.2.1.8 SAADC active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{SAADC,RUN}$	SAADC sampling @ 16 ksps, Acquisition time = 20 µs, Clock = HFXO, Regulator = DCDC		1.1		mA

5.2.1.9 COMP active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{COMP,LP}$	COMP enabled, low power mode		17.2		µA
$I_{COMP,NORM}$	COMP enabled, normal mode		21		µA
$I_{COMP,HS}$	COMP enabled, high-speed mode		28.7		µA

5.2.1.10 WDT active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{WDT,STARTED}$	WDT started		1.3		µA

5.3 POWER — Power supply

This device has the following power supply features:

- On-chip LDO and DC/DC regulators
- Global System ON/OFF modes with individual RAM section power control
- Analog or digital pin wakeup from System OFF
- Supervisor HW to manage power on reset, brownout, and power fail
- Auto-controlled refresh modes for LDO and DC/DC regulators to maximize efficiency
- Automatic switching between LDO and DC/DC regulator based on load to maximize efficiency

Note: Two additional external passive components are required to use the DC/DC regulator.

5.3.1 Regulators

The following internal power regulator alternatives are supported:

- Internal LDO regulator
- Internal DC/DC regulator

The LDO is the default regulator.

The DC/DC regulator can be used as an alternative to the LDO regulator and is enabled through the DCDCEN on page 70 register. Using the DC/DC regulator will reduce current consumption compared to when using the LDO regulator, but the DC/DC regulator requires an external LC filter to be connected, as shown in DC/DC regulator setup on page 62.

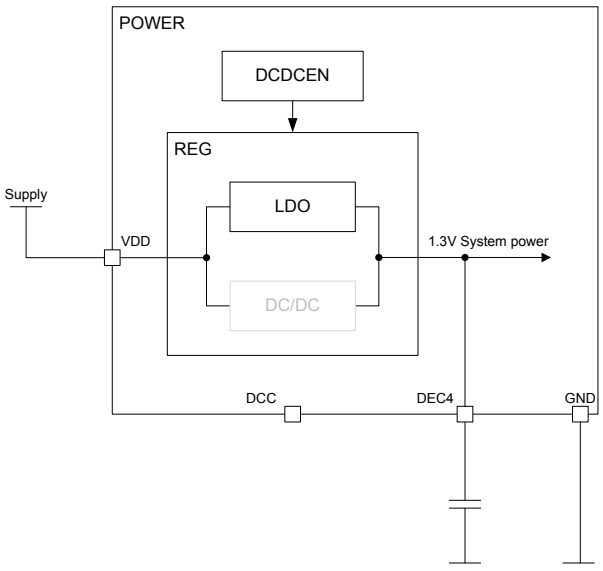

Figure 13: LDO regulator setup

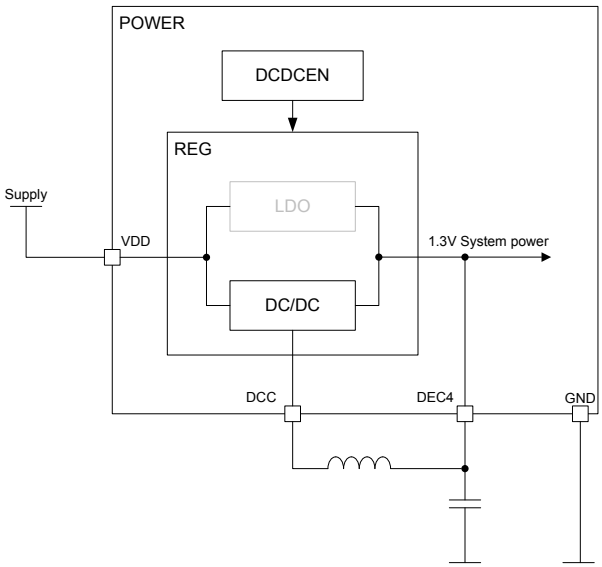

Figure 14: DC/DC regulator setup

5.3.2 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:

1. The DETECT signal, optionally generated by the GPIO peripheral
2. A reset

When the system wakes up from System OFF mode, it gets reset. For more details, see Reset behavior on page 66.

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, see Reset behavior. Note that these registers are usually overwritten by the startup code provided with the nRF application examples.

Before entering System OFF mode, the user must make sure that all on-going EasyDMA transactions have been completed. This is usually accomplished by making sure that the EasyDMA enabled peripheral is not active when entering System OFF.

5.3.2.1 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 on page 50 for more information. Required resources needed for debugging include the following key components: Debug on page 50, CLOCK — Clock control on page 83, POWER — Power supply on page 61, NVMC — Non-volatile memory controller on page 18, CPU, 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.

5.3.3 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 68 provides information about the source that caused the wakeup or reset.

The system can switch on and off the appropriate internal power sources, 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.

5.3.3.1 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 will be constant and kept at a minimum. This is secured by forcing a set of base resources on while in sleep. The advantage of having a constant and predictable latency will be at the 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 63, ensures the most efficient supply option is chosen to save the most power. The advantage of having the lowest power possible will be at the cost of having 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 will, by default, reside in the low power sub-power mode.

5.3.4 Power supply supervisor

The power supply supervisor initializes the system at power-on and provides an early warning of impending power failure.

In addition, the power supply supervisor puts the system in a reset state if the supply voltage is too low for safe operation (brownout). The power supply supervisor is illustrated in Power supply supervisor on page 64.

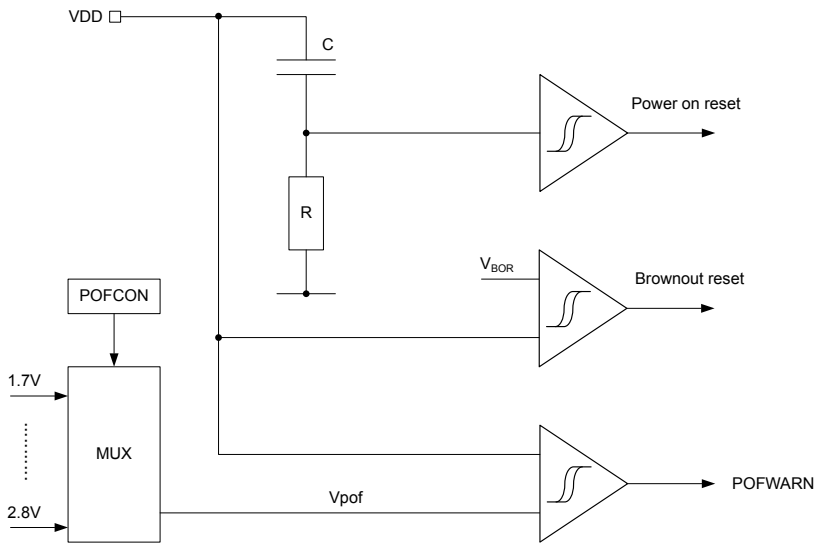

Figure 15: Power supply supervisor

5.3.4.1 Power-fail comparator

The power-fail comparator (POF) can provide the CPU with an early warning of impending power failure. It will not reset the system, but give the CPU time to prepare for an orderly power-down.

The comparator features a hysteresis of V_{HYST}, as illustrated in Power-fail comparator (BOR = Brownout reset) on page 64. The threshold V_{POF} is set in register POFCON on page 69. If the POF is enabled and the supply voltage falls below V_{POF}, the POFWARN event will be generated. This event will also be generated if the supply voltage is already below V_{POF} at the time the POF is enabled, or if V_{POF} is re-configured to a level above the supply voltage.

If power-fail warning is enabled and the supply voltage is below V_{POF} the power-fail comparator will prevent the NVMC from performing write operations to the NVM. See NVMC — Non-volatile memory controller on page 18 for more information about the NVMC.

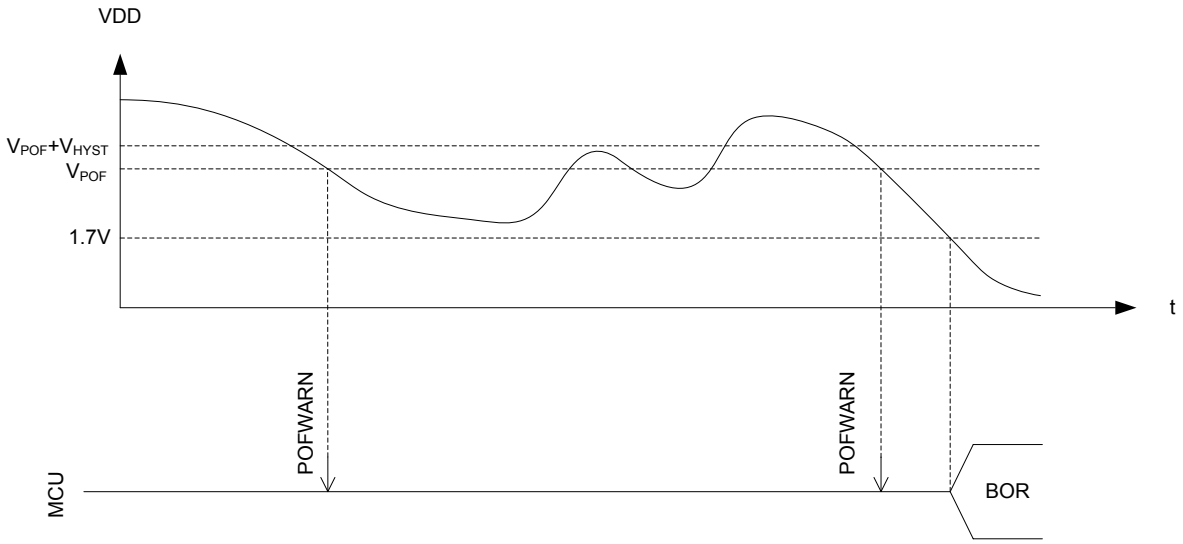

Figure 16: Power-fail comparator (BOR = Brownout reset)

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

5.3.7 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 individual peripheral chapters for information of which registers are retained for the various peripherals.

5.3.8 Reset behavior

Reset source	Reset target								
	CPU	Peripherals	GPIO	Debug[a]	SWJ-DP	RAM	WDT	Retained registers	RESETREAS
CPU lockup [5]	x	x	x						
Soft reset	x	x	x						
Wakeup from System OFF mode reset	x	x		x [6]		x [7]			
Watchdog reset [8]	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 reset source, RAM content may be corrupted.

5.3.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	POWER	POWER	Power control	For 24 kB RAM variant, only RAM[0].x to RAM[2].x registers are in use.

Table 14: Instances

Register	Offset	Description
TASKS_CONSTLAT	0x078	Enable constant latency mode
TASKS_LOWPWR	0x07C	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
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RESETREAS	0x400	Reset reason
SYSTEMOFF	0x500	System OFF register
POFCON	0x510	Power failure comparator configuration
GPREGRET	0x51C	General purpose retention register

[a] All debug components excluding SWJ-DP. See Debug on page 50 chapter for more information about the different debug components in the system.

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

[6] The Debug components will not be reset if the device is in debug interface mode.

[7] RAM is not reset on wakeup from OFF mode, but depending on settings in the RAM register parts, or the whole RAM, may not be retained after the device has entered System OFF mode.

[8] Watchdog reset is not available in System OFF.

Register	Offset	Description
GPREGRET2	0x520	General purpose retention register
DCDCEN	0x578	DC/DC enable register
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

Table 15: Register Overview

5.3.9.1 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	POFWARN			Write '1' to Enable interrupt for POFWARN event See EVENTS_POFWARN
			Set	1	Enable
			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

Bit number				31 30 29 28 27 26 25 24 23 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

5.3.9.2 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	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
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

5.3.9.3 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				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	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

Bit number				31 30 29 28 27 26 25 24 23 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
			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
			Detected	1	Detected
F	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

5.3.9.4 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

5.3.9.5 POFCON

Address offset: 0x510

Power failure 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				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 comparator
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	THRESHOLD			Power failure comparator threshold setting
			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
			V27	14	Set threshold to 2.7 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				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
			V28	15	Set threshold to 2.8 V

5.3.9.6 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

5.3.9.7 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

5.3.9.8 DCDCEN

Address offset: 0x578

DC/DC 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	DCDCEN			Enable or disable DC/DC converter
			Disabled	0	Disable
			Enabled	1	Enable

5.3.9.9 RAM[0].POWER

Address offset: 0x900

RAM0 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.10 RAM[0].POWERSET

Address offset: 0x904

RAM0 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				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	W	S0POWER			Keep RAM section S0 of RAM0 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM0 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.11 RAM[0].POWERCLR

Address offset: 0x908

RAM0 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				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	W	S0POWER			Keep RAM section S0 of RAM0 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM0 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.12 RAM[1].POWER

Address offset: 0x910

RAM1 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.13 RAM[1].POWERSET

Address offset: 0x914

RAM1 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM1 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM1 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.14 RAM[1].POWERCLR

Address offset: 0x918

RAM1 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM1 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM1 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.15 RAM[2].POWER

Address offset: 0x920

RAM2 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.16 RAM[2].POWERSET

Address offset: 0x924

RAM2 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				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	W	S0POWER			Keep RAM section S0 of RAM2 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM2 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.17 RAM[2].POWERCLR

Address offset: 0x928

RAM2 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				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	W	S0POWER			Keep RAM section S0 of RAM2 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM2 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.18 RAM[3].POWER

Address offset: 0x930

RAM3 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.19 RAM[3].POWERSET

Address offset: 0x934

RAM3 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM3 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM3 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.20 RAM[3].POWERCLR

Address offset: 0x938

RAM3 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM3 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM3 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.21 RAM[4].POWER

Address offset: 0x940

RAM4 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.22 RAM[4].POWERSET

Address offset: 0x944

RAM4 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				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	W	S0POWER			Keep RAM section S0 of RAM4 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM4 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.23 RAM[4].POWERCLR

Address offset: 0x948

RAM4 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				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	W	S0POWER			Keep RAM section S0 of RAM4 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM4 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.24 RAM[5].POWER

Address offset: 0x950

RAM5 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.25 RAM[5].POWERSET

Address offset: 0x954

RAM5 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM5 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM5 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.26 RAM[5].POWERCLR

Address offset: 0x958

RAM5 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				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**
Id	RW	Field	Value Id	Value	Description
A	W	S0POWER			Keep RAM section S0 of RAM5 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM5 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.27 RAM[6].POWER

Address offset: 0x960

RAM6 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.28 RAM[6].POWERSET

Address offset: 0x964

RAM6 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				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	W	S0POWER			Keep RAM section S0 of RAM6 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM6 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.29 RAM[6].POWERCLR

Address offset: 0x968

RAM6 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				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	W	S0POWER			Keep RAM section S0 of RAM6 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM6 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.9.30 RAM[7].POWER

Address offset: 0x970

RAM7 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				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	RW	S0POWER			Keep RAM section S0 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S0RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
B	RW	S1POWER			Keep RAM section S1 ON or OFF in System ON mode. RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in S1RETENTION. All RAM sections will be OFF in System OFF mode.
			Off	0	Off
			On	1	On
C	RW	S0RETENTION			Keep retention on RAM section S0 when RAM section is in OFF
			Off	0	Off
			On	1	On
D	RW	S1RETENTION			Keep retention on RAM section S1 when RAM section is in OFF
			Off	0	Off
			On	1	On

5.3.9.31 RAM[7].POWERSET

Address offset: 0x974

RAM7 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				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	W	S0POWER			Keep RAM section S0 of RAM7 on or off in System ON mode
			On	1	On
B	W	S1POWER			Keep RAM section S1 of RAM7 on or off in System ON mode
			On	1	On
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			On	1	On
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			On	1	On

5.3.9.32 RAM[7].POWERCLR

Address offset: 0x978

RAM7 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				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	W	S0POWER			Keep RAM section S0 of RAM7 on or off in System ON mode
			Off	1	Off
B	W	S1POWER			Keep RAM section S1 of RAM7 on or off in System ON mode
			Off	1	Off
C	W	S0RETENTION			Keep retention on RAM section S0 when RAM section is switched off
			Off	1	Off
D	W	S1RETENTION			Keep retention on RAM section S1 when RAM section is switched off
			Off	1	Off

5.3.10 Electrical specification

5.3.10.1 Device startup times

Symbol	Description	Min.	Typ.	Max.	Units
t_{POR}	Time in Power on Reset after VDD reaches 1.7 V for all supply voltages and temperatures. Dependent on supply rise time. [9]				
$t_{POR,10us}$	VDD rise time 10us		1		ms

[9] A step increase in supply voltage of 300 mV or more, with rise time of 300 ms or less, within the valid supply range, may result in a system reset.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{POR,10ms}$	VDD rise time 10ms		9		ms
$t_{POR,60ms}$	VDD rise time 60ms		23		ms
t_{PINR}	If a GPIO pin is configured as reset, the maximum time taken to pull up the pin and release reset after power on reset. Dependent on the pin capacitive load (C)[10]: t=5RC, R = 13kOhm				
$t_{PINR,500nF}$	C = 500nF			32.5	ms
$t_{PINR,10uF}$	C = 10uF			650	ms
t_{R2ON}	Time from reset to ON (CPU execute)				
$t_{R2ON,NOTCONF}$	If reset pin not configured	tPOR			ms
$t_{R2ON,CONF}$	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

5.3.10.2 Power fail comparator

Symbol	Description	Min.	Typ.	Max.	Units
V_{POF}	Nominal power level warning thresholds (falling supply voltage). Levels are configurable between Min. and Max. in 100mV increments.	1.7		2.8	V
V_{POFTOL}	Threshold voltage tolerance		±1	±5	%
$V_{POFHYST}$	Threshold voltage hysteresis		50		mV
$V_{BOR,OFF}$	Brown out reset voltage range SYSTEM OFF mode	1.2		1.7	V
$V_{BOR,ON}$	Brown out reset voltage range SYSTEM ON mode	1.48		1.7	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 oscillator activity for low latency start up
- Automatic oscillator and clock control, and distribution for ultra-low power

[10] To decrease maximum time a device could hold in reset, a strong external pullup resistor can be used.

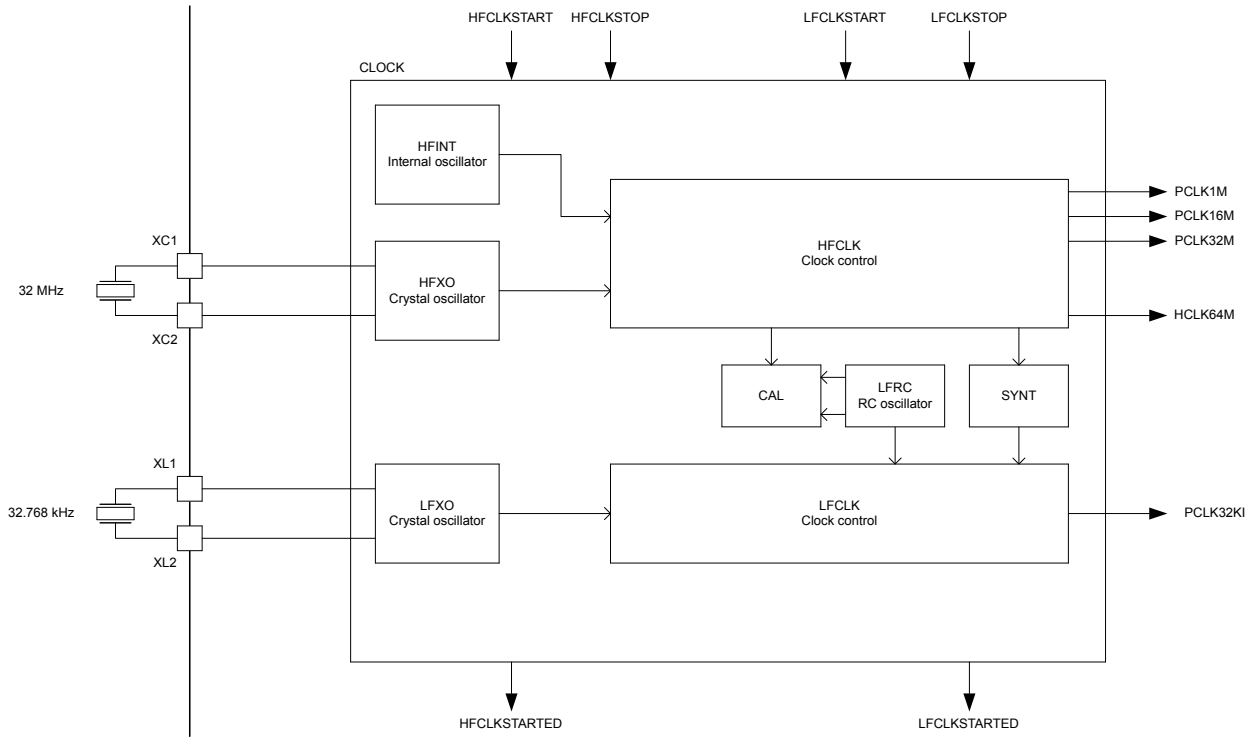

Figure 17: Clock control

5.4.1 HFCLK clock controller

The HFCLK clock controller provides the following clocks to the system.

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

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

- 64 MHz internal oscillator (HFINT)
- 64 MHz crystal oscillator (HFXO)

For illustration, see Clock control on page 84.

When the system requests one or more clocks from the HFCLK controller, the HFCLK controller will automatically provide them. If the system does not request any clocks provided by the HFCLK controller, the controller will enter a power saving mode.

These clocks are only available when the system is in ON mode. When the system enters ON mode, the internal oscillator (HFINT) clock source will automatically start to be able to provide the required HFCLK clock(s) for the system.

The HFINT will be used when HFCLK is requested and HFXO has not been started. The HFXO is started by triggering the HFCLKSTART task and stopped using the HFCLKSTOP task. A HFCLKSTARTED event will be generated when the HFXO has started and its frequency is stable.

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 LFCLK clock is stopped by triggering the LFCLKSTOP task.

It is not allowed to write to register LFCLKSRC on page 91 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 90 indicates a 'LFCLK running' state.

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

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 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. See Table 32.768 kHz RC oscillator (LFRC) on page 92 for details on the default and calibrated accuracy of the LFRC oscillator.
The LFRC oscillator does not require additional external components.

5.4.2.2 Calibrating the 32.768 kHz RC oscillator

After the 32.768 kHz RC oscillator is started and running, it can be calibrated by triggering the CAL task. In this case, the HFCLK will be temporarily switched on and used as a reference.

A DONE event will be generated when calibration has finished. The calibration mechanism will only work as long as HFCLK is generated from the HFCLK crystal oscillator, it is therefore necessary to explicitly start this crystal oscillator before calibration can be started, see HFCLKSTART task.

It is not allowed to stop the LFRC 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 and generate a CTTO timeout event when it reaches 0. The Calibration timer will stop by itself when it reaches 0.

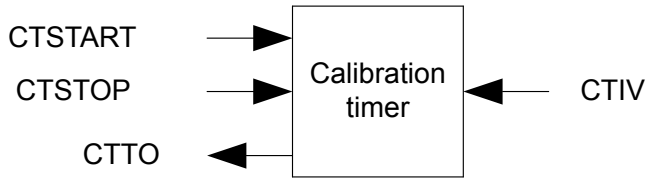

Figure 19: Calibration timer

Due to limitations in the calibration timer, only one task related to calibration, that is, CAL, CTSTART and CTSTOP, can be triggered for every period of LFCLK.

5.4.2.4 32.768 kHz crystal oscillator (LFXO)

For higher LFCLK accuracy 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.

The LFCLKSRC on page 91 register controls the clock source, and its allowed swing. The truth table for various situations is as follows:

5.4.3 Registers

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

Table 17: Instances

Register	Offset	Description	
TASKS_HFCLKSTART	0x000	Start HFCLK crystal oscillator	
TASKS_HFCLKSTOP	0x004	Stop HFCLK crystal oscillator	
TASKS_LFCLKSTART	0x008	Start LFCLK source	
TASKS_LFCLKSTOP	0x00C	Stop LFCLK source	
TASKS_CAL	0x010	Start calibration of LFRC oscillator	
TASKS_CTSTART	0x014	Start calibration timer	
TASKS_CTSTOP	0x018	Stop calibration timer	
EVENTS_HFCLKSTARTED	0x100	HFCLK oscillator started	
EVENTS_LFCLKSTARTED	0x104	LFCLK started	
EVENTS_DONE	0x10C	Calibration of LFCLK RC oscillator complete event	
EVENTS_CTTO	0x110	Calibration timer timeout	
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	
CTIV	0x538	Calibration timer interval	Retained

Table 18: 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				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

Bit number				31 30 29 28 27 26 25 24 23 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	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

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				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

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
			Xtal	1	32.768 kHz crystal oscillator
			Synth	2	32.768 kHz synthesized from HFCLK
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
			Xtal	1	32.768 kHz crystal oscillator
			Synth	2	32.768 kHz synthesized from HFCLK

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
			Xtal	1	32.768 kHz crystal oscillator
			Synth	2	32.768 kHz synthesized from HFCLK
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 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

5.4.4 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	%
t_{START_HFINT}	Startup time		3		us

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			±40	ppm
C_{L_HFXO}	Load capacitance			12	pF
C_{0_HFXO}	Shunt capacitance			7	pF
$R_{S\_HFXO\_7PF}$	Equivalent series resistance C0 = 7 pF			60	ohm
$R_{S\_HFXO\_5PF}$	Equivalent series resistance C0 = 5 pF			60	ohm
$R_{S\_HFXO\_3PF}$	Equivalent series resistance C0 = 3 pF			100	ohm
P_{D_HFXO}	Drive level			100	uW
C_{PIN_HFXO}	Input capacitance XC1 and XC2		4		pF
t_{START_HFXO}	Startup time		0.36		ms

5.4.4.3 32.768 kHz RC oscillator (LFRC)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFRC}	Nominal frequency		32.768		kHz
f_{TOL_LFRC}	Frequency tolerance			±2	%
$f_{TOL\_CAL\_LFRC}$	Frequency tolerance for LFRC after calibration[11]			±500	ppm
t_{START_LFRC}	Startup time for 32.768 kHz RC oscillator		600		us

5.4.4.4 32.768 kHz crystal oscillator (LFXO)

[11] Constant temperature within ±0.5 °C and calibration performed at least every 8 seconds

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFXO}	Crystal frequency		32.768		kHz
$f_{TOL\_LFXO\_BLE}$	Frequency tolerance requirement for BLE stack			±250	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	kohm
P_{D_LFXO}	Drive level			1	uW
C_{pin}	Input capacitance on XL1 and XL2 pads		4		pF
t_{START_LFXO}	Startup time for 32.768 kHz crystal oscillator		0.25		s
$V_{AMP\_IN\_XO\_LOW}$	Peak to peak amplitude for external low swing clock. Input signal must not swing outside supply rails.	200		1000	mV

5.4.4.5 32.768 kHz synthesized from HFCLK (LFSYNT)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFSYNT}	Nominal frequency		32.768		kHz
f_{TOL_LFSYNT}	Frequency tolerance in addition to HFLCK tolerance[12]		8		ppm
t_{START_LFSYNT}	Startup time for synthesized 32.768 kHz		100		us

[12] Frequency tolerance will be derived from the HFCLK source clock plus the LFSYNT tolerance

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.

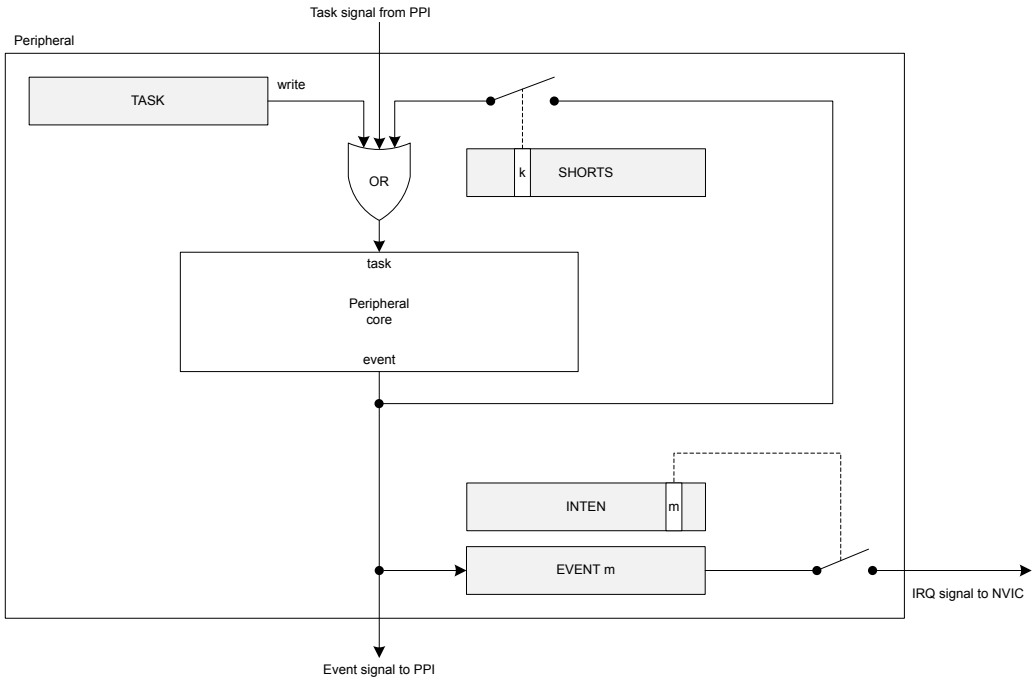

Figure 21: Tasks, events, shortcuts, and interrupts

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 17 for more information about which peripherals are available and where they are located in the address map.

There is a direct relationship between the 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, and with the exception of ID 0, peripherals sharing an ID and base address may not be used simultaneously. The user can only enable one at the time on this specific ID.

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

- Disable the previously used peripheral
- Remove any 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 enable and do not rely on configuration values that may be inherited from the peripheral that was disabled.
- Enable the now configured peripheral.

See Instantiation on page 17 to see which peripherals are sharing ID.

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 a dedicated SET and CLR register in that 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 a '1' to a bit in the SET or CLR register will set or clear the same bit in the main register respectively. Writing a '0' to a bit in the SET or CLR register has no effect. Reading the SET or CLR registers returns the value of the main register.

Restriction: 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 a '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 94.

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 94. An event register is only cleared when firmware writes a '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, its 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, and the INTEN register is not available on those peripherals. Refer to the individual chapters for details. In all cases, however, 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 94.

Interrupt clearing

When clearing an interrupt by writing "0" to an event register, or disabling an interrupt using the INTENCLR register, it 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.

> **Important:** 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 clear or interrupt disable another 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 101, ADDRPTR on page 101 and the SCRATCHPTR on page 101 is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 15 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.

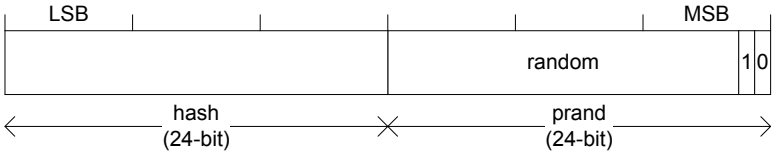

Figure 22: Resolvable address

To resolve an address the ADDRPTR on page 101 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[13]. 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.

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

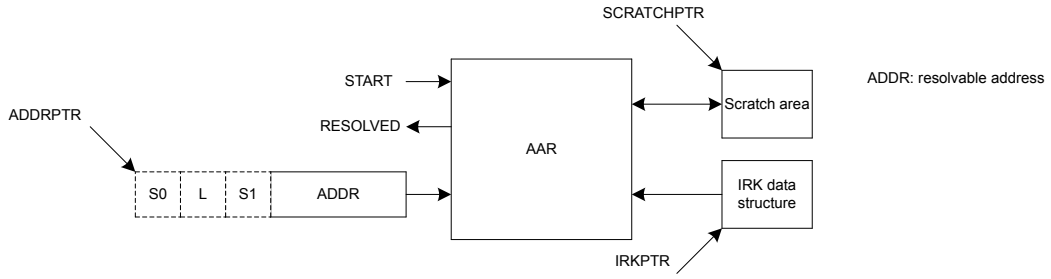

Figure 23: Address resolution with packet preloaded into RAM

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.

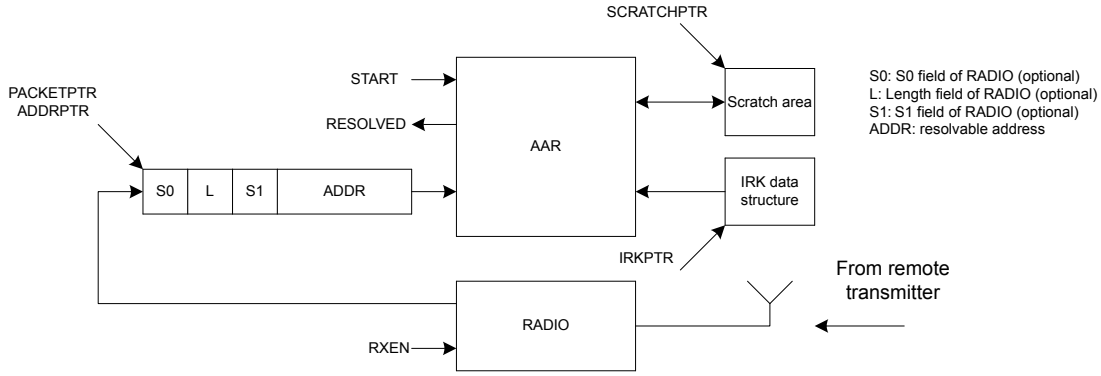

Figure 24: Address resolution with packet loaded into RAM by the RADIO

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 19: IRK data structure overview

6.2.5 Registers

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

Table 20: 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 21: 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).	..	..	..	µs
$t_{AAR,8}$	Time for address resolution of 8 IRKs. (Given priority to the actual destination RAM block).		48		µs

6.3 BPROT — Block protection

The mechanism for protecting non-volatile memory can be used to prevent erroneous application code from erasing or writing to protected blocks.

Non-volatile memory can be protected from erases and writes depending on the settings in the CONFIG registers. One bit in a CONFIG register represents one protected block of 4 kB. There are multiple CONFIG registers to cover the whole range of the flash. Protected regions of program memory on page 102 illustrates how the CONFIG bits map to the program memory space.

Important: If an erase or write to a protected block is detected, the CPU will hard fault. If an ERASEALL operation is attempted from the CPU while any block is protected it will be blocked and the CPU will hard fault.

On reset, all the protection bits are cleared. To ensure safe operation, the first task after reset must be to set the protection bits. The only way of clearing protection bits is by resetting the device from any reset source.

The protection mechanism is turned off when in debug mode (a debugger is connected) and the DISABLEINDEBUG register is set to disable.

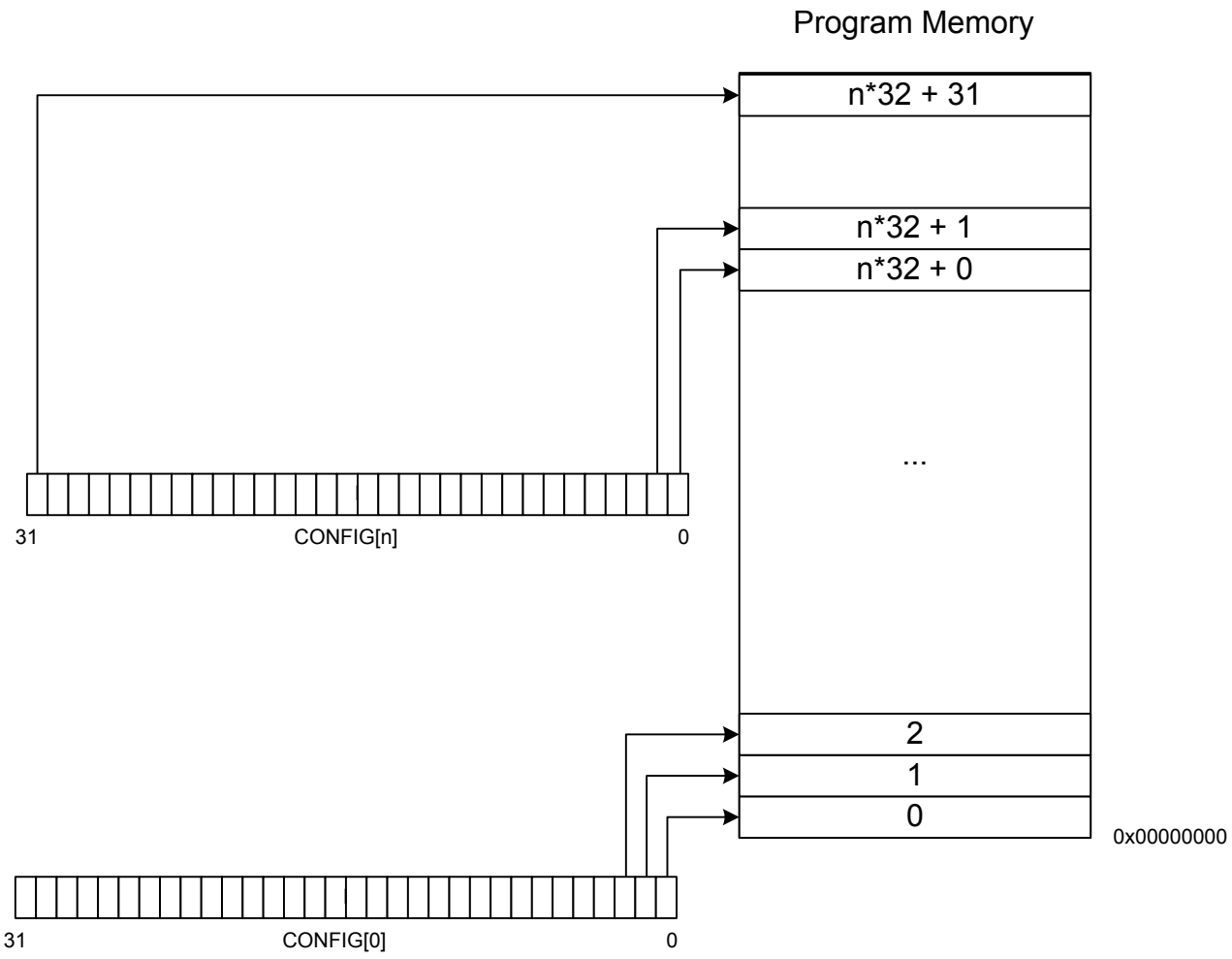

Figure 25: Protected regions of program memory

6.3.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	BPROT	BPROT	Block protect	

Table 22: Instances

Register	Offset	Description	
CONFIG0	0x600	Block protect configuration register 0	
CONFIG1	0x604	Block protect configuration register 1	
DISABLEINDEBUG	0x608	Disable protection mechanism in debug mode	
	0x60C		Reserved

Table 23: Register Overview

6.3.1.1 CONFIG0

Address offset: 0x600

Block protect 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	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	RW	REGION0			Enable protection for region 0. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
B	RW	REGION1			Enable protection for region 1. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
C	RW	REGION2			Enable protection for region 2. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
D	RW	REGION3			Enable protection for region 3. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
E	RW	REGION4			Enable protection for region 4. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
F	RW	REGION5			Enable protection for region 5. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
G	RW	REGION6			Enable protection for region 6. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
H	RW	REGION7			Enable protection for region 7. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
I	RW	REGION8			Enable protection for region 8. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
J	RW	REGION9			Enable protection for region 9. Write '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				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
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
K	RW	REGION10			Enable protection for region 10. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
L	RW	REGION11			Enable protection for region 11. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
M	RW	REGION12			Enable protection for region 12. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
N	RW	REGION13			Enable protection for region 13. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
O	RW	REGION14			Enable protection for region 14. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
P	RW	REGION15			Enable protection for region 15. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
Q	RW	REGION16			Enable protection for region 16. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
R	RW	REGION17			Enable protection for region 17. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
S	RW	REGION18			Enable protection for region 18. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
T	RW	REGION19			Enable protection for region 19. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
U	RW	REGION20			Enable protection for region 20. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
V	RW	REGION21			Enable protection for region 21. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
W	RW	REGION22			Enable protection for region 22. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
X	RW	REGION23			Enable protection for region 23. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
Y	RW	REGION24			Enable protection for region 24. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
Z	RW	REGION25			Enable protection for region 25. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection 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 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	RW	REGION26			Enable protection for region 26. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
b	RW	REGION27			Enable protection for region 27. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
c	RW	REGION28			Enable protection for region 28. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
d	RW	REGION29			Enable protection for region 29. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
e	RW	REGION30			Enable protection for region 30. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
f	RW	REGION31			Enable protection for region 31. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled

6.3.1.2 CONFIG1

Address offset: 0x604

Block protect 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				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	REGION32			Enable protection for region 32. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
B	RW	REGION33			Enable protection for region 33. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
C	RW	REGION34			Enable protection for region 34. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
D	RW	REGION35			Enable protection for region 35. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
E	RW	REGION36			Enable protection for region 36. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
F	RW	REGION37			Enable protection for region 37. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
G	RW	REGION38			Enable protection for region 38. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection 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
H	RW	REGION39			Enable protection for region 39. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
I	RW	REGION40			Enable protection for region 40. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
J	RW	REGION41			Enable protection for region 41. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
K	RW	REGION42			Enable protection for region 42. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
L	RW	REGION43			Enable protection for region 43. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
M	RW	REGION44			Enable protection for region 44. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
N	RW	REGION45			Enable protection for region 45. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
O	RW	REGION46			Enable protection for region 46. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled
P	RW	REGION47			Enable protection for region 47. Write '0' has no effect.
			Disabled	0	Protection disabled
			Enabled	1	Protection enabled

6.3.1.3 DISABLEINDEBUG

Address offset: 0x608

Disable protection mechanism in debug 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 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	DISABLEINDEBUG			Disable the protection mechanism for NVM regions while in debug mode. This register will only disable the protection mechanism if the device is in debug mode.
			Disabled	1	Disabled in debug
			Enabled	0	Enabled in debug

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

Figure 134: Top layer

Figure 135: Bottom layer

Important: 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	Supply voltage, independent of DCDC enable		1.7	3.0	3.6	V
t_{R_VDD}	Supply rise time (0 V to 1.7 V)				60	ms
TA	Operating temperature		-40	25	85	°C

Table 107: 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
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
Radio				
RF input level			10	dBm
Environmental (QFN package)				
Storage temperature		-40	+125	°C
MSL	Moisture Sensitivity Level		2	
ESD HBM	Human Body Model		4	kV
ESD CDM	Charged Device Model		1000	V
Flash memory				
Endurance		10 000		Write/erase cycles
Retention		10 years at 40°C		

Table 108: Absolute maximum ratings

10 Ordering information

This chapter contains information on IC marking, ordering codes, and container sizes.

10.1 IC marking

The IC package is marked like described below.

N	5	2	8	1	0
<P	P>	<V	V>	<H>	<P>
<Y	Y>	<W	W>	<L	L>

Figure 136: Package marking

10.2 Box labels

Here are the box labels used for the IC.

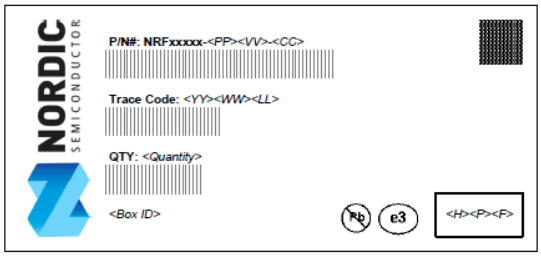

Figure 137: Inner box label

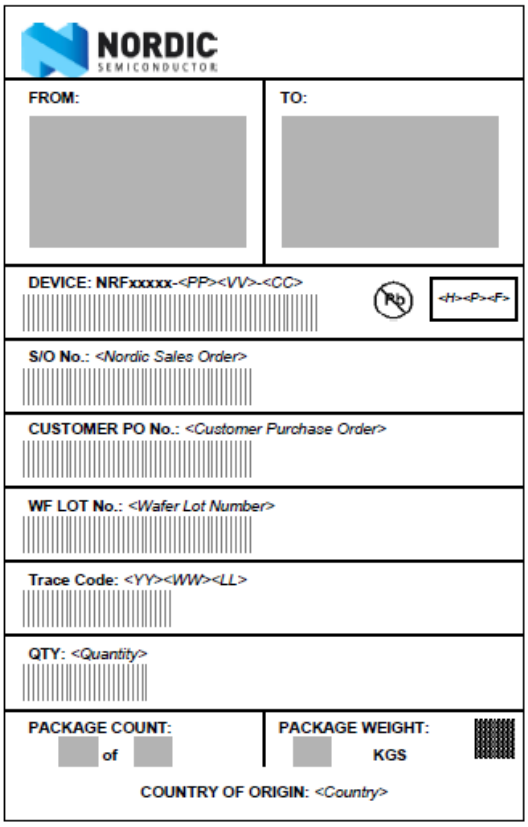

Figure 138: Outer box label

10.3 Order code

Here are the nRF52810 order codes and definitions.

n	R	F	5	2	8	1	0	-	<P	P>	<V	V>	-	<C	C>

Figure 139: Order code

Abbrevitation	Definition and implemented codes
N52/nRF52	nRF52 Series product
810	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 109: Abbreviations

10.4 Code ranges and values

Defined here are the nRF52810 code ranges and values.

<PP>	Package	Size (mm)	Pin/Ball count	Pitch (mm)
QF	QFN	6 x 6	48	0.4
QC	QFN	5 x 5	32	0.5

Table 110: Package variant codes

<VV>	Flash (kB)	RAM (kB)
AA	192	24

Table 111: Function variant codes

<H>	Description
[A . . Z]	Hardware version/revision identifier (incremental)

Table 112: Hardware version codes

<P>	Description
[0 . . 9]	Production device identifier (incremental)
[A . . Z]	Engineering device identifier (incremental)

Table 113: Production configuration codes

<F>	Description
[A . . N, P . . Z]	Version of preprogrammed firmware
[0]	Delivered without preprogrammed firmware

Table 114: Production version codes

<YY>	Description
[15 . . 99]	Production year: 2015 to 2099

Table 115: Year codes

<WW>	Description
[1 . . 52]	Week of production

Table 116: Week codes

<LL>	Description
[AA . . ZZ]	Wafer production lot identifier

Table 117: Lot codes

<CC>	Description
R7	7" Reel
R	13" Reel
T	Tray

Table 118: Container codes

10.5 Product options

Defined here are the nRF52810 product options.

Order code	MOQ (minimum ordering quantity)	Comment
nRF52810-QFAA-R7	1000	Availability to be announced.
nRF52810-QFAA-R	3000	
nRF52810-QFAA-T	490	
nRF52810-QCAA-R7	1000	
nRF52810-QCAA-R	3000	
nRF52810-QCAA-T	490	

Table 119: nRF IC order codes

Order code	Description
nRF52-DK	nRF52832 development kit with tools to support nRF52810 development.

Table 120: Development tools order code

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.

Life support applications

Nordic Semiconductor's 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.

RoHS and REACH statement

Nordic Semiconductor's 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 www.nordicsemi.com.