



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



nRF52810

Product Specification

v1.0

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 μ A/MHz running from flash memory
 - 32.8 μ A/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 μ A at 3 V in System OFF mode, no RAM retention
 - 0.5 μ A at 3 V in System OFF mode with full 24 kB RAM retention
 - 1.5 μ A 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 ksp/s 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

5.3.1	Regulators	61
5.3.2	System OFF mode	62
5.3.3	System ON mode	63
5.3.4	Power supply supervisor	63
5.3.5	RAM sections	65
5.3.6	Reset	65
5.3.7	Retained registers	66
5.3.8	Reset behavior	66
5.3.9	Registers	66
5.3.10	Electrical specification	82
5.4	CLOCK — Clock control	83
5.4.1	HFCLK clock controller	84
5.4.2	LFCLK clock controller	85
5.4.3	Registers	88
5.4.4	Electrical specification	92
6	Peripherals	94
6.1	Peripheral interface	94
6.1.1	Peripheral ID	94
6.1.2	Peripherals with shared ID	95
6.1.3	Peripheral registers	95
6.1.4	Bit set and clear	95
6.1.5	Tasks	95
6.1.6	Events	95
6.1.7	Shortcuts	96
6.1.8	Interrupts	96
6.2	AAR — Accelerated address resolver	97
6.2.1	EasyDMA	97
6.2.2	Resolving a resolvable address	97
6.2.3	Use case example for chaining RADIO packet reception with address resolution using AAR	98
6.2.4	IRK data structure	98
6.2.5	Registers	98
6.2.6	Electrical specification	101
6.3	BPROT — Block protection	101
6.3.1	Registers	103
6.4	CCM — AES CCM mode encryption	106
6.4.1	Key-stream generation	107
6.4.2	Encryption	107
6.4.3	Decryption	108
6.4.4	AES CCM and RADIO concurrent operation	108
6.4.5	Encrypting packets on-the-fly in radio transmit mode	109
6.4.6	Decrypting packets on-the-fly in radio receive mode	110
6.4.7	CCM data structure	111
6.4.8	EasyDMA and ERROR event	112
6.4.9	Registers	112
6.4.10	Electrical specification	117
6.5	COMP — Comparator	117
6.5.1	Differential mode	119
6.5.2	Single-ended mode	119
6.5.3	Registers	121
6.5.4	Electrical specification	127
6.6	ECB — AES electronic codebook mode encryption	127
6.6.1	Shared resources	128
6.6.2	EasyDMA	128

6.6.3	ECB data structure	128
6.6.4	Registers	128
6.6.5	Electrical specification	130
6.7	EGU — Event generator unit	130
6.7.1	Registers	130
6.7.2	Electrical specification	137
6.8	GPIO — General purpose input/output	137
6.8.1	Pin configuration	137
6.8.2	Registers	139
6.8.3	Electrical specification	187
6.9	GPIOE — GPIO tasks and events	188
6.9.1	Pin events and tasks	189
6.9.2	Port event	189
6.9.3	Tasks and events pin configuration	190
6.9.4	Registers	190
6.9.5	Electrical specification	201
6.10	PDM — Pulse density modulation interface	201
6.10.1	Master clock generator	202
6.10.2	Module operation	202
6.10.3	Decimation filter	202
6.10.4	EasyDMA	203
6.10.5	Hardware example	204
6.10.6	Pin configuration	204
6.10.7	Registers	205
6.10.8	Electrical specification	210
6.11	PPI — Programmable peripheral interconnect	210
6.11.1	Pre-programmed channels	212
6.11.2	Registers	212
6.12	PWM — Pulse width modulation	252
6.12.1	Wave counter	252
6.12.2	Decoder with EasyDMA	256
6.12.3	Limitations	263
6.12.4	Pin configuration	263
6.12.5	Registers	264
6.12.6	Electrical specification	273
6.13	QDEC — Quadrature decoder	273
6.13.1	Sampling and decoding	274
6.13.2	LED output	275
6.13.3	Debounce filters	275
6.13.4	Accumulators	275
6.13.5	Output/input pins	276
6.13.6	Pin configuration	276
6.13.7	Registers	277
6.13.8	Electrical specification	284
6.14	RADIO — 2.4 GHz radio	285
6.14.1	EasyDMA	285
6.14.2	Packet configuration	286
6.14.3	Maximum packet length	287
6.14.4	Address configuration	287
6.14.5	Data whitening	287
6.14.6	CRC	288
6.14.7	Radio states	289
6.14.8	Transmit sequence	289
6.14.9	Receive sequence	291

6.14.10	Received signal strength indicator (RSSI)	292
6.14.11	Interframe spacing	292
6.14.12	Device address match	293
6.14.13	Bit counter	293
6.14.14	Registers	294
6.14.15	Electrical specification	313
6.15	RNG — Random number generator	318
6.15.1	Bias correction	319
6.15.2	Speed	319
6.15.3	Registers	319
6.15.4	Electrical specification	321
6.16	RTC — Real-time counter	321
6.16.1	Clock source	321
6.16.2	Resolution versus overflow and the PRESCALER	322
6.16.3	COUNTER register	322
6.16.4	Overflow features	323
6.16.5	TICK event	323
6.16.6	Event control feature	323
6.16.7	Compare feature	324
6.16.8	TASK and EVENT jitter/delay	326
6.16.9	Reading the COUNTER register	328
6.16.10	Registers	329
6.16.11	Electrical specification	335
6.17	SAADC — Successive approximation analog-to-digital converter	335
6.17.1	Shared resources	335
6.17.2	Overview	335
6.17.3	Digital output	336
6.17.4	Analog inputs and channels	337
6.17.5	Operation modes	337
6.17.6	EasyDMA	339
6.17.7	Resistor ladder	340
6.17.8	Reference	341
6.17.9	Acquisition time	341
6.17.10	Limits event monitoring	342
6.17.11	Registers	343
6.17.12	Electrical specification	370
6.17.13	Performance factors	371
6.18	SPIM — Serial peripheral interface master with EasyDMA	371
6.18.1	SPI master transaction sequence	372
6.18.2	Master mode pin configuration	373
6.18.3	EasyDMA	373
6.18.4	Low power	375
6.18.5	Registers	375
6.18.6	Electrical specification	381
6.19	SPIS — Serial peripheral interface slave with EasyDMA	382
6.19.1	Shared resources	383
6.19.2	EasyDMA	383
6.19.3	SPI slave operation	384
6.19.4	Pin configuration	385
6.19.5	Registers	386
6.19.6	Electrical specification	395
6.20	SWI — Software interrupts	397
6.20.1	Registers	397
6.21	TEMP — Temperature sensor	397

6.21.1	Registers	397
6.21.2	Electrical specification	403
6.22	TIMER — Timer/counter	403
6.22.1	Capture	404
6.22.2	Compare	404
6.22.3	Task delays	404
6.22.4	Task priority	404
6.22.5	Registers	405
6.23	TWIM — I ² C compatible two-wire interface master with EasyDMA	411
6.23.1	EasyDMA	412
6.23.2	Master write sequence	412
6.23.3	Master read sequence	413
6.23.4	Master repeated start sequence	414
6.23.5	Low power	415
6.23.6	Master mode pin configuration	415
6.23.7	Registers	416
6.23.8	Electrical specification	424
6.23.9	Pullup resistor	425
6.24	TWIS — I ² C compatible two-wire interface slave with EasyDMA	425
6.24.1	EasyDMA	427
6.24.2	TWI slave responding to a read command	427
6.24.3	TWI slave responding to a write command	428
6.24.4	Master repeated start sequence	430
6.24.5	Terminating an ongoing TWI transaction	430
6.24.6	Low power	431
6.24.7	Slave mode pin configuration	431
6.24.8	Registers	431
6.24.9	Electrical specification	439
6.25	UARTE — Universal asynchronous receiver/transmitter with EasyDMA	439
6.25.1	EasyDMA	440
6.25.2	Transmission	440
6.25.3	Reception	441
6.25.4	Error conditions	443
6.25.5	Using the UARTE without flow control	443
6.25.6	Parity and stop bit configuration	443
6.25.7	Low power	443
6.25.8	Pin configuration	444
6.25.9	Registers	444
6.25.10	Electrical specification	454
6.26	WDT — Watchdog timer	454
6.26.1	Reload criteria	454
6.26.2	Temporarily pausing the watchdog	454
6.26.3	Watchdog reset	454
6.26.4	Registers	455
6.26.5	Electrical specification	460
7	Hardware and layout	461
7.1	Pin assignments	461
7.1.1	QFN48 pin assignments	461
7.1.2	QFN32 pin assignments	463
7.1.3	GPIO pins located near the radio	466
7.2	Mechanical specifications	466
7.2.1	QFN48 6 x 6 mm package	466
7.2.2	QFN32 5 x 5 mm package	467

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 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0																														
Id	RW	Field	Value Id	Value	Description																										
A	RW	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 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0																															
Id	RW	Field	Value Id	Value	Description																												
B	RW	FIELD_B			Example of a deprecated field																												
			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																												

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.

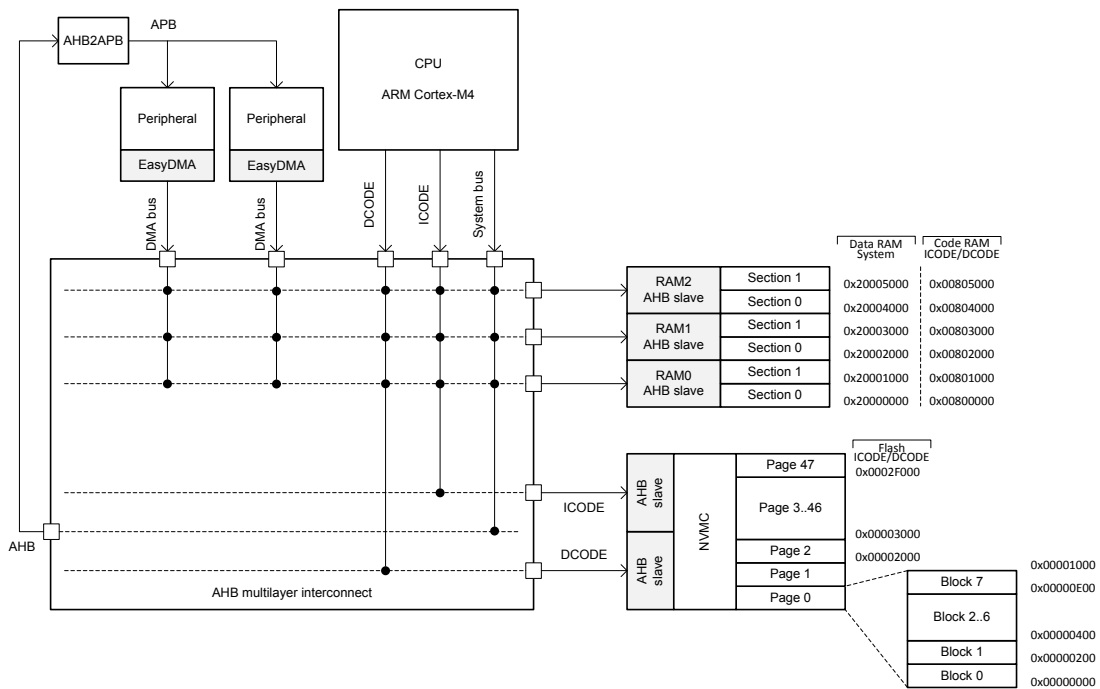


Figure 2: Memory layout

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.

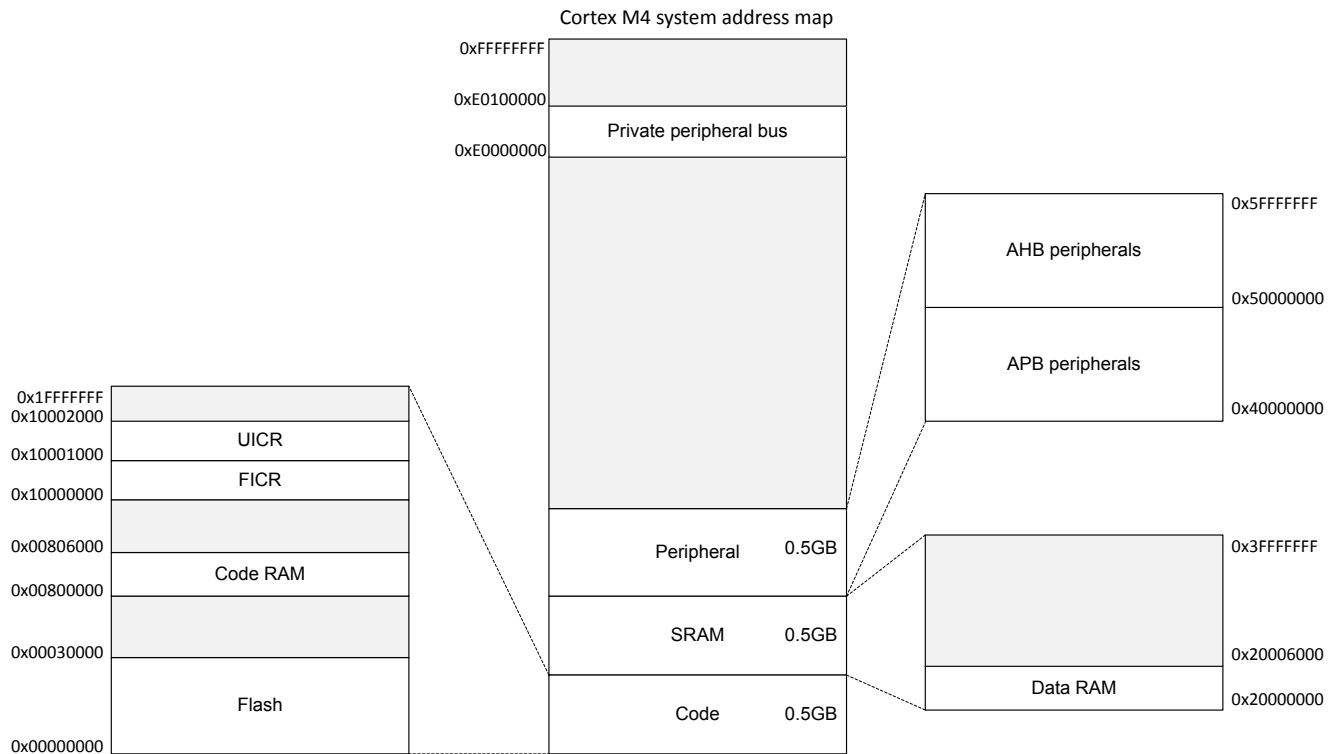


Figure 3: Memory map

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 $t_{\text{ERASEPAGE}}$. 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	
<code>READY</code>	0x400	Ready flag	
<code>CONFIG</code>	0x504	Configuration register	
<code>ERASEPCR1</code>	0x508	Register for erasing a page in code area. Equivalent to <code>ERASEPAGE</code> .	Deprecated
<code>ERASEPAGE</code>	0x508	Register for erasing a page in code area	
<code>ERASEALL</code>	0x50C	Register for erasing all non-volatile user memory	
<code>ERASEPCRO</code>	0x510	Register for erasing a page in code area. Equivalent to <code>ERASEPAGE</code> .	Deprecated
<code>ERASEUICR</code>	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																														
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																															
Reset 0x00000000	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 ERASEPCRO (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																															
Reset 0x00000000	0 0																															
Id	RW	Field	Value Id	Value	Description																											
A	RW	ERASEPCRO			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																															
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				μ s
$t_{ERASEPAGE}$	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	A	A
Reset 0x00001000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Id	RW	Field	Value	Id	Value	Description																										
A	R	CODEPAGESIZE				Code memory page size																										

4.4.1.2 CODESIZE

Address offset: 0x014

Code memory size

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