



# nRF52 Product Specification v1.2

Key features	Applications
<ul style="list-style-type: none"><li>2.4 GHz transceiver<ul style="list-style-type: none"><li>-96 dBm sensitivity in Bluetooth® low energy mode</li><li>1 Mbps, 2 Mbps supported data rates</li><li>TX power -20 to +4 dBm in 4 dB steps</li><li>Single-pin antenna interface</li><li>5.3 mA peak current in TX (0 dBm)</li><li>5.4 mA peak current in RX</li><li>RSSI (1 dB resolution)</li></ul></li><li>ARM® Cortex®-M4 32-bit processor with FPU, 64 MHz<ul style="list-style-type: none"><li>215 EEMBC CoreMark® score running from flash memory</li><li>58 µA/MHz running from flash memory</li><li>51.6 µA/MHz running from RAM</li><li>Data watchpoint and trace (DWT), embedded trace macrocell (ETM), and instrumentation trace macrocell (ITM)</li><li>Serial wire debug (SWD)</li><li>Trace port</li></ul></li><li>Flexible power management<ul style="list-style-type: none"><li>Supply voltage range 1.7 V–3.6 V</li><li>Fully automatic LDO and DC/DC regulator system</li><li>Fast wake-up using 64 MHz internal oscillator</li><li>0.3 µA at 3 V in OFF mode</li><li>0.7 µA at 3 V in OFF mode with full 64 kB RAM retention</li><li>1.9 µA at 3 V in ON mode, no RAM retention, wake on RTC</li></ul></li><li>Memory<ul style="list-style-type: none"><li>512 kB flash/64 kB RAM</li><li>256 kB flash/32 kB RAM</li></ul></li><li>Nordic SoftDevice ready</li><li>Support for concurrent multi-protocol</li><li>Type 2 near field communication (NFC-A) tag with wakeup-on-field and touch-to-pair capabilities</li><li>12-bit, 200 ksps ADC - 8 configurable channels with programmable gain</li><li>64 level comparator</li><li>15 level low power comparator with wakeup from System OFF mode</li><li>Temperature sensor</li><li>32 general purpose I/O pins</li><li>3x 4-channel pulse width modulator (PWM) units with EasyDMA</li><li>Digital microphone interface (PDM)</li><li>5x 32-bit timers with counter mode</li><li>Up to 3x SPI master/slave with EasyDMA</li><li>Up to 2x I2C compatible 2-Wire master/slave</li><li>I2S with EasyDMA</li><li>UART (CTS/RTS) with EasyDMA</li><li>Programmable peripheral interconnect (PPI)</li><li>Quadrature decoder (QDEC)</li><li>AES HW encryption with EasyDMA</li><li>Autonomous peripheral operation without CPU intervention using PPI and EasyDMA</li><li>3x real-time counter (RTC)</li><li>External system<ul style="list-style-type: none"><li>Single crystal operation</li><li>On-chip balun (single-ended RF)</li><li>Few external components</li></ul></li><li>Package variants<ul style="list-style-type: none"><li>QFN48 package, 6 × 6 mm</li><li>WLCSP package, 3.0 × 3.2 mm</li></ul></li></ul>	<ul style="list-style-type: none"><li>Internet of Things (IoT)<ul style="list-style-type: none"><li>Home automation</li><li>Sensor networks</li><li>Building automation</li><li>Industrial</li><li>Retail</li></ul></li><li>Personal area networks<ul style="list-style-type: none"><li>Health/fitness sensor and monitor devices</li><li>Medical devices</li><li>Key fobs and wrist watches</li></ul></li><li>Interactive entertainment devices<ul style="list-style-type: none"><li>Remote controls</li><li>Gaming controllers</li></ul></li><li>Beacons</li><li>A4WP wireless chargers and devices</li><li>Remote control toys</li><li>Computer peripherals and I/O devices<ul style="list-style-type: none"><li>Mouse</li><li>Keyboard</li><li>Multi-touch trackpad</li><li>Gaming</li></ul></li></ul>

All rights reserved.

Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

2016-09-29

# Contents

<b>1 Revision history.....</b>	<b>9</b>
<b>2 About this document.....</b>	<b>10</b>
2.1 Peripheral naming and abbreviations.....	10
2.2 Register tables.....	10
2.3 Registers.....	11
<b>3 Block diagram.....</b>	<b>12</b>
<b>4 Pin assignments.....</b>	<b>13</b>
4.1 QFN48 pin assignments.....	13
4.2 WLCSP ball assignments.....	15
4.3 GPIO usage restrictions.....	17
<b>5 Absolute maximum ratings.....</b>	<b>19</b>
<b>6 Recommended operating conditions.....</b>	<b>20</b>
<b>7 CPU.....</b>	<b>21</b>
7.1 Floating point interrupt.....	21
7.2 Electrical specification.....	21
7.3 CPU and support module configuration.....	22
<b>8 Memory.....</b>	<b>23</b>
8.1 RAM - Random access memory.....	23
8.2 Flash - Non-volatile memory.....	24
8.3 Memory map.....	24
8.4 Instantiation.....	24
<b>9 AHB multilayer.....</b>	<b>26</b>
9.1 AHB multilayer priorities.....	26
<b>10 EasyDMA.....</b>	<b>27</b>
10.1 EasyDMA array list.....	28
<b>11 NVMC — Non-volatile memory controller.....</b>	<b>29</b>
11.1 Writing to Flash.....	29
11.2 Erasing a page in Flash.....	29
11.3 Writing to user information configuration registers (UICR).....	29
11.4 Erasing user information configuration registers (UICR).....	29
11.5 Erase all.....	30
11.6 Cache.....	30
11.7 Registers.....	30
11.8 Electrical specification.....	33
<b>12 BPROT — Block protection.....</b>	<b>34</b>
12.1 Registers.....	34
<b>13 FICR — Factory information configuration registers.....</b>	<b>43</b>
13.1 Registers.....	43
<b>14 UICR — User information configuration registers.....</b>	<b>54</b>
14.1 Registers.....	54
<b>15 Peripheral interface.....</b>	<b>68</b>
15.1 Peripheral ID.....	68
15.2 Peripherals with shared ID.....	68
15.3 Peripheral registers.....	69
15.4 Bit set and clear.....	69
15.5 Tasks.....	69
15.6 Events.....	70
15.7 Shortcuts.....	70
15.8 Interrupts.....	70

<b>16 Debug and trace.....</b>	<b>72</b>
16.1 DAP - Debug Access Port.....	72
16.2 CTRL-AP - Control Access Port.....	73
16.3 Debug interface mode.....	74
16.4 Real-time debug.....	74
16.5 Trace.....	75
<b>17 Power and clock management.....</b>	<b>76</b>
17.1 Current consumption scenarios.....	76
<b>18 POWER — Power supply.....</b>	<b>78</b>
18.1 Regulators.....	78
18.2 System OFF mode.....	79
18.3 System ON mode.....	80
18.4 Power supply supervisor.....	80
18.5 RAM sections.....	82
18.6 Reset.....	82
18.7 Retained registers.....	83
18.8 Reset behavior.....	83
18.9 Registers.....	83
18.10 Electrical specification.....	99
<b>19 CLOCK — Clock control.....</b>	<b>101</b>
19.1 HFCLK clock controller.....	101
19.2 LFCLK clock controller.....	103
19.3 Registers.....	105
19.4 Electrical specification.....	109
<b>20 GPIO — General purpose input/output.....</b>	<b>111</b>
20.1 Pin configuration.....	111
20.2 GPIO located near the RADIO.....	113
20.3 Registers.....	113
20.4 Electrical specification.....	154
<b>21 GPIOTE — GPIO tasks and events.....</b>	<b>157</b>
21.1 Pin events and tasks.....	157
21.2 Port event.....	158
21.3 Tasks and events pin configuration.....	158
21.4 Registers.....	158
21.5 Electrical specification.....	167
<b>22 PPI — Programmable peripheral interconnect.....</b>	<b>168</b>
22.1 Pre-programmed channels.....	169
22.2 Registers.....	169
<b>23 RADIO — 2.4 GHz Radio.....</b>	<b>204</b>
23.1 EasyDMA.....	204
23.2 Packet configuration.....	205
23.3 Maximum packet length.....	206
23.4 Address configuration.....	206
23.5 Data whitening.....	206
23.6 CRC.....	207
23.7 Radio states.....	208
23.8 Transmit sequence.....	208
23.9 Receive sequence.....	210
23.10 Received Signal Strength Indicator (RSSI).....	211
23.11 Interframe spacing.....	211
23.12 Device address match.....	212
23.13 Bit counter.....	212
23.14 Registers.....	213
23.15 Electrical specification.....	229
<b>24 TIMER — Timer/counter.....</b>	<b>233</b>
24.1 Capture.....	234
24.2 Compare.....	234

24.3 Task delays.....	234
24.4 Task priority.....	234
24.5 Registers.....	234
24.6 Electrical specification.....	240
<b>25 RTC — Real-time counter.....</b>	<b>241</b>
25.1 Clock source.....	241
25.2 Resolution versus overflow and the PRESCALER.....	241
25.3 COUNTER register.....	242
25.4 Overflow features.....	242
25.5 TICK event.....	242
25.6 Event control feature.....	243
25.7 Compare feature.....	243
25.8 TASK and EVENT jitter/delay.....	245
25.9 Reading the COUNTER register.....	247
25.10 Registers.....	247
25.11 Electrical specification.....	253
<b>26 RNG — Random number generator.....</b>	<b>254</b>
26.1 Bias correction.....	254
26.2 Speed.....	254
26.3 Registers.....	254
26.4 Electrical specification.....	256
<b>27 TEMP — Temperature sensor.....</b>	<b>257</b>
27.1 Registers.....	257
27.2 Electrical specification.....	262
<b>28 ECB — AES electronic codebook mode encryption.....</b>	<b>263</b>
28.1 Shared resources.....	263
28.2 EasyDMA.....	263
28.3 ECB data structure.....	263
28.4 Registers.....	264
28.5 Electrical specification.....	265
<b>29 CCM — AES CCM mode encryption.....</b>	<b>266</b>
29.1 Shared resources.....	267
29.2 Encryption.....	267
29.3 Decryption.....	267
29.4 AES CCM and RADIO concurrent operation.....	268
29.5 Encrypting packets on-the-fly in radio transmit mode.....	268
29.6 Decrypting packets on-the-fly in radio receive mode.....	269
29.7 CCM data structure.....	270
29.8 EasyDMA and ERROR event.....	271
29.9 Registers.....	271
<b>30 AAR — Accelerated address resolver.....</b>	<b>275</b>
30.1 Shared resources.....	275
30.2 EasyDMA.....	275
30.3 Resolving a resolvable address.....	275
30.4 Use case example for chaining RADIO packet reception with address resolution using AAR.....	276
30.5 IRK data structure.....	276
30.6 Registers.....	277
30.7 Electrical specification.....	279
<b>31 SPIM — Serial peripheral interface master with EasyDMA.....</b>	<b>280</b>
31.1 Shared resources.....	280
31.2 EasyDMA.....	281
31.3 SPI master transaction sequence.....	282
31.4 Low power.....	283
31.5 Master mode pin configuration.....	283
31.6 Registers.....	284
31.7 Electrical specification.....	289

---

<b>32 SPIS — Serial peripheral interface slave with EasyDMA.....</b>	<b>291</b>
32.1 Shared resources.....	291
32.2 EasyDMA.....	291
32.3 SPI slave operation.....	292
32.4 Pin configuration.....	293
32.5 Registers.....	294
32.6 Electrical specification.....	302
<b>33 TWIM — I<sup>2</sup>C compatible two-wire interface master with EasyDMA.....</b>	<b>304</b>
33.1 Shared resources.....	305
33.2 EasyDMA.....	305
33.3 Master write sequence.....	306
33.4 Master read sequence.....	307
33.5 Master repeated start sequence.....	308
33.6 Low power.....	309
33.7 Master mode pin configuration.....	309
33.8 Registers.....	309
33.9 Electrical specification.....	316
<b>34 TWIS — I<sup>2</sup>C compatible two-wire interface slave with EasyDMA.....</b>	<b>318</b>
34.1 Shared resources.....	320
34.2 EasyDMA.....	320
34.3 TWI slave responding to a read command.....	320
34.4 TWI slave responding to a write command.....	321
34.5 Master repeated start sequence.....	322
34.6 Terminating an ongoing TWI transaction.....	323
34.7 Low power.....	323
34.8 Slave mode pin configuration.....	323
34.9 Registers.....	324
34.10 Electrical specification.....	330
<b>35 UARTE — Universal asynchronous receiver/transmitter with EasyDMA....</b>	<b>332</b>
35.1 Shared resources.....	332
35.2 EasyDMA.....	332
35.3 Transmission.....	333
35.4 Reception.....	333
35.5 Error conditions.....	335
35.6 Using the UARTE without flow control.....	335
35.7 Parity configuration.....	335
35.8 Low power.....	335
35.9 Pin configuration.....	336
35.10 Registers.....	336
35.11 Electrical specification.....	344
<b>36 QDEC — Quadrature decoder.....</b>	<b>346</b>
36.1 Sampling and decoding.....	346
36.2 LED output.....	347
36.3 Debounce filters.....	347
36.4 Accumulators.....	348
36.5 Output/input pins.....	348
36.6 Pin configuration.....	348
36.7 Registers.....	349
36.8 Electrical specification.....	355
<b>37 SAADC — Successive approximation analog-to-digital converter.....</b>	<b>356</b>
37.1 Shared resources.....	356
37.2 Overview.....	356
37.3 Digital output.....	357
37.4 Analog inputs and channels.....	358
37.5 Operation modes.....	358
37.6 EasyDMA.....	360
37.7 Resistor ladder.....	361

37.8 Reference.....	362
37.9 Acquisition time.....	362
37.10 Limits event monitoring.....	363
37.11 Registers.....	364
37.12 Electrical specification.....	388
37.13 Performance factors.....	390
<b>38 COMP — Comparator.....</b>	<b>391</b>
38.1 Shared resources.....	392
38.2 Differential mode.....	392
38.3 Single-ended mode.....	393
38.4 Pin configuration.....	395
38.5 Registers.....	395
38.6 Electrical specification.....	400
<b>39 LPCOMP — Low power comparator.....</b>	<b>402</b>
39.1 Shared resources.....	403
39.2 Pin configuration.....	403
39.3 Registers.....	404
39.4 Electrical specification.....	408
<b>40 WDT — Watchdog timer.....</b>	<b>409</b>
40.1 Reload criteria.....	409
40.2 Temporarily pausing the watchdog.....	409
40.3 Watchdog reset.....	409
40.4 Registers.....	410
40.5 Electrical specification.....	414
<b>41 SWI — Software interrupts.....</b>	<b>415</b>
41.1 Registers.....	415
<b>42 NFCT — Near field communication tag.....</b>	<b>416</b>
42.1 Overview.....	416
42.2 Pin configuration.....	418
42.3 EasyDMA.....	418
42.4 Collision resolution.....	419
42.5 Frame timing controller.....	420
42.6 Frame assembler.....	421
42.7 Frame disassembler.....	422
42.8 Antenna interface.....	423
42.9 NFCT antenna recommendations.....	423
42.10 Battery protection.....	423
42.11 References.....	424
42.12 Registers.....	424
42.13 Electrical specification.....	435
<b>43 PDM — Pulse density modulation interface.....</b>	<b>436</b>
43.1 Master clock generator.....	436
43.2 Module operation.....	436
43.3 Decimation filter.....	437
43.4 EasyDMA.....	437
43.5 Hardware example.....	438
43.6 Pin configuration.....	438
43.7 Registers.....	439
43.8 Electrical specification.....	443
<b>44 I<sup>2</sup>S — Inter-IC sound interface.....</b>	<b>445</b>
44.1 Mode.....	445
44.2 Transmitting and receiving.....	445
44.3 Left right clock (LRCK).....	446
44.4 Serial clock (SCK).....	446
44.5 Master clock (MCK).....	447
44.6 Width, alignment and format.....	447
44.7 EasyDMA.....	449

44.8 Module operation.....	451
44.9 Pin configuration.....	452
44.10 Registers.....	453
44.11 Electrical specification.....	460
<b>45 MWU — Memory watch unit.....</b>	<b>461</b>
45.1 Registers.....	461
<b>46 EGU — Event generator unit.....</b>	<b>488</b>
46.1 Registers.....	488
46.2 Electrical specification.....	494
<b>47 PWM — Pulse width modulation.....</b>	<b>495</b>
47.1 Wave counter.....	495
47.2 Decoder with EasyDMA.....	498
47.3 Limitations.....	503
47.4 Pin configuration.....	503
47.5 Registers.....	504
47.6 Electrical specification.....	512
<b>48 SPI — Serial peripheral interface master.....</b>	<b>513</b>
48.1 Functional description.....	513
48.2 Registers.....	516
48.3 Electrical specification.....	519
<b>49 TWI — I<sup>2</sup>C compatible two-wire interface.....</b>	<b>521</b>
49.1 Functional description.....	521
49.2 Master mode pin configuration.....	521
49.3 Shared resources.....	522
49.4 Master write sequence.....	522
49.5 Master read sequence.....	523
49.6 Master repeated start sequence.....	524
49.7 Low power.....	525
49.8 Registers.....	525
49.9 Electrical specification.....	529
<b>50 UART — Universal asynchronous receiver/transmitter.....</b>	<b>531</b>
50.1 Functional description.....	531
50.2 Pin configuration.....	531
50.3 Shared resources.....	532
50.4 Transmission.....	532
50.5 Reception.....	532
50.6 Suspending the UART.....	533
50.7 Error conditions.....	533
50.8 Using the UART without flow control.....	534
50.9 Parity configuration.....	534
50.10 Registers.....	534
50.11 Electrical specification.....	539
<b>51 Mechanical specifications.....</b>	<b>540</b>
51.1 QFN48 6 x 6 mm package.....	540
51.2 WLCSP package.....	541
<b>52 Ordering information.....</b>	<b>542</b>
52.1 IC marking.....	542
52.2 Box labels.....	542
52.3 Order code.....	543
52.4 Code ranges and values.....	543
52.5 Product options.....	544
<b>53 Reference circuitry.....</b>	<b>545</b>
53.1 Schematic QFAA and QFAB QFN48 with internal LDO setup.....	545
53.2 Schematic QFAA and QFAB QFN48 with DC/DC regulator setup.....	546
53.3 Schematic QFAA and QFAB QFN48 with DC/DC regulator and NFC setup.....	547
53.4 Schematic CIAA WLCSP with internal LDO setup.....	548
53.5 Schematic CIAA WLCSP with DC/DC regulator setup.....	549

53.6 Schematic CIAA WLCSP with DC/DC regulator and NFC setup.....	550
53.7 PCB guidelines.....	550
53.8 PCB layout example.....	551
<b>54 Liability disclaimer.....</b>	<b>553</b>
54.1 Life support applications.....	553

# 1 Revision history

Date	Version	Description
September 2016	1.2	<p>Updated the following:</p> <ul style="list-style-type: none"> <li>• Power and clock management, <a href="#">Current consumption: Ultra-low power</a> on page 77.</li> <li>• Power, <a href="#">Current consumption, sleep</a> on page 99</li> </ul>
July 2016	1.1	<p>Added documentation for nRF52832 CIAA WLCSP.</p> <p>Added or updated the following content:</p> <ul style="list-style-type: none"> <li>• Cover: Added <a href="#">Key features</a>.</li> <li>• <a href="#">Pin assignments</a> on page 13: Added <a href="#">WLCSP ball assignments</a>. Moved <a href="#">GPIO usage restrictions</a> here from <a href="#">GPIO/Notes on usage and restrictions</a>.</li> <li>• <a href="#">Absolute maximum ratings</a> on page 19: Added environmental information for WLCSP to the table.</li> <li>• <a href="#">Memory</a> on page 23: Added QFAB and CIAA information to the table.</li> <li>• <a href="#">FICR — Factory information configuration registers</a> on page 43: Updated INFO.PACKAGE register.</li> <li>• <a href="#">UICR</a>: Updated APPROTECT register.</li> <li>• <a href="#">Debug and trace</a> on page 72: Updated DAP - Debug access port.</li> <li>• <a href="#">POWER — Power supply</a> on page 78: Updated Pin reset.</li> <li>• <a href="#">CLOCK — Clock control</a> on page 101: Updated information on external 32 kHz clock support.</li> <li>• <a href="#">GPIO — General purpose input/output</a> on page 111: Added <a href="#">GPIO located near the RADIO</a>.</li> <li>• <a href="#">RADIO — 2.4 GHz Radio</a> on page 204: Updated Figure 29 and <a href="#">Interframe spacing</a>.</li> <li>• <a href="#">CCM</a>: Updated SCRATCHPTR register.</li> <li>• <a href="#">SPIM</a>: Updated <a href="#">Master mode pin configuration</a>.</li> <li>• <a href="#">UARTE</a>: Added RXDRDY and TXDRDY events.</li> <li>• <a href="#">NFCT</a>: Updated <a href="#">Electrical specifications</a>.</li> <li>• <a href="#">PWM — Pulse width modulation</a> on page 495: Updated SEQ[1].REFRESH register.</li> <li>• <a href="#">Mechanical specifications</a> on page 540: Added <a href="#">WLCSP package</a>.</li> <li>• <a href="#">Ordering information</a> on page 542: Updated with CIAA and QFAB information.</li> <li>• <a href="#">Reference circuitry</a> on page 545: QFAB information added. CIAA WLCSP schematics added.</li> </ul>
February 2016	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 broken into separate sections that include the following information:

- A detailed functional description of the peripheral.
- Register configuration for the peripheral
- Electrical specification tables providing the specified limits of the chip when tested under the conditions defined in the *Recommended operating conditions* on page 20

### 2.1 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.

### 2.2 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.

#### 2.2.1 Fields and values

The **Id (Field Id)** row specifies the bits that belong to the different fields in the register.

A blank space means that the field is reserved and that it is read as undefined, and 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.

If a field has enumerated values, then every value will be identified with a unique value id in the **Value Id** column. Single-bit bit fields may, however, omit the **Value Id** when values can be substituted with a Boolean type enumerator range, for example, True/False, Disable/Enable, On/Off, and so on.

Values are usually provided as decimal or hexadecimal. Hex 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], that is, 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 “..”.

When an item is marked with the word **Deprecated**, it means this is an attribute applied to a feature to indicate that it should not be used for new designs.

## 2.3 Registers

**Table 1: Register Overview**

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

### **2.3.1 DUMMY**

Address offset: 0x514

Example of a register controlling a dummy feature

## 3 Block diagram

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

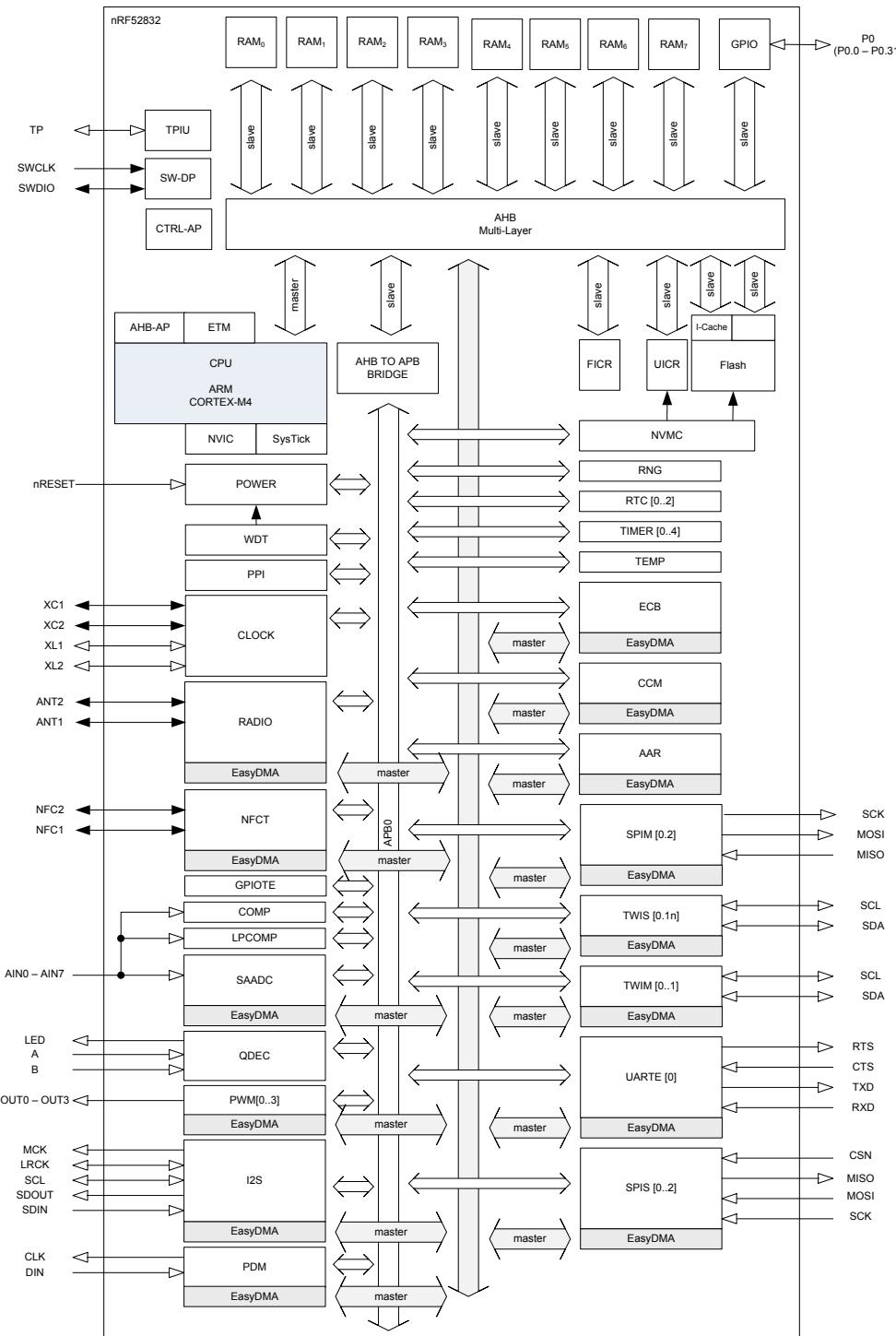


Figure 1: Block diagram

# 4 Pin assignments

Here we cover the pin assignments for each variant of the chip.

## 4.1 QFN48 pin assignments

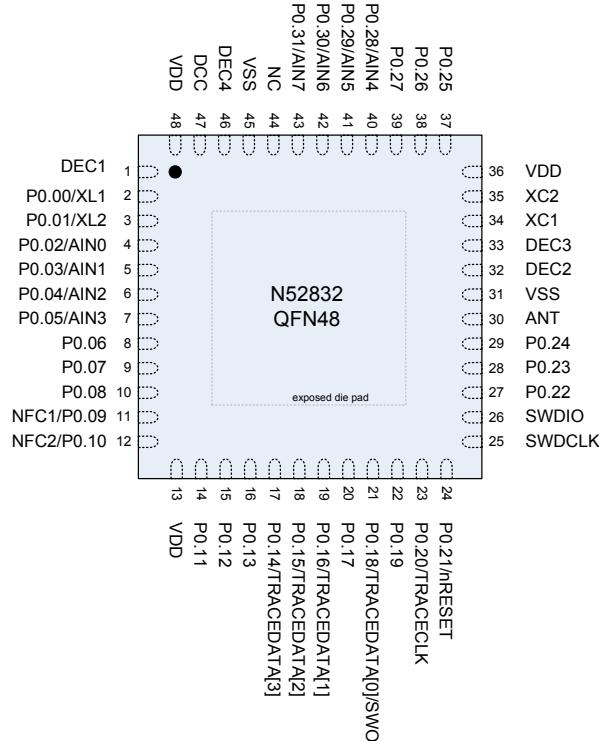


Figure 2: QFN48 pin assignments, top view

Table 2: QFN48 pin assignments

Pin	Name	Type	Description
<b>Left Side of chip</b>			
1	DEC1	Power	0.9 V regulator digital supply decoupling
2	P0.00	Digital I/O	General purpose I/O
	XL1	Analog input	Connection for 32.768 kHz crystal (LFXO)
3	P0.01	Digital I/O	General purpose I/O
	XL2	Analog input	Connection for 32.768 kHz crystal (LFXO)
4	P0.02	Digital I/O	General purpose I/O
	AIN0	Analog input	SAADC/COMP/LPCOMP input
5	P0.03	Digital I/O	General purpose I/O
	AIN1	Analog input	SAADC/COMP/LPCOMP input
6	P0.04	Digital I/O	General purpose I/O
	AIN2	Analog input	SAADC/COMP/LPCOMP input
7	P0.05	Digital I/O	General purpose I/O
	AIN3	Analog input	SAADC/COMP/LPCOMP input
8	P0.06	Digital I/O	General purpose I/O
9	P0.07	Digital I/O	General purpose I/O

Pin	Name	Type	Description
10	P0.08	Digital I/O	General purpose I/O
11	NFC1	NFC input	NFC antenna connection
	P0.09	Digital I/O	General purpose I/O <sup>1</sup>
12	NFC2	NFC input	NFC antenna connection
	P0.10	Digital I/O	General purpose I/O <sup>1</sup>
<b>Bottom side of chip</b>			
13	VDD	Power	Power supply
14	P0.11	Digital I/O	General purpose I/O
15	P0.12	Digital I/O	General purpose I/O
16	P0.13	Digital I/O	General purpose I/O
17	P0.14	Digital I/O	General purpose I/O
	TRACEDATA[3]		Trace port output
18	P0.15	Digital I/O	General purpose I/O
	TRACEDATA[2]		Trace port output
19	P0.16	Digital I/O	General purpose I/O
	TRACEDATA[1]		Trace port output
20	P0.17	Digital I/O	General purpose I/O
21	P0.18	Digital I/O	General purpose I/O
	TRACEDATA[0] / SWO		Single wire output
			Trace port output
22	P0.19	Digital I/O	General purpose I/O
23	P0.20	Digital I/O	General purpose I/O
	TRACECLK		Trace port clock output
24	P0.21	Digital I/O	General purpose I/O
	nRESET		Configurable as pin reset
<b>Right Side of chip</b>			
25	SWDCLK	Digital input	Serial wire debug clock input for debug and programming
26	SWDIO	Digital I/O	Serial wire debug I/O for debug and programming
27	P0.22	Digital I/O	General purpose I/O <sup>2</sup>
28	P0.23	Digital I/O	General purpose I/O <sup>2</sup>
29	P0.24	Digital I/O	General purpose I/O <sup>2</sup>
30	ANT	RF	Single-ended radio antenna connection
31	VSS	Power	Ground (Radio supply)
32	DEC2	Power	1.3 V regulator supply decoupling (Radio supply)
33	DEC3	Power	Power supply decoupling
34	XC1	Analog input	Connection for 32 MHz crystal
35	XC2	Analog input	Connection for 32 MHz crystal
36	VDD	Power	Power supply
<b>Top side of chip</b>			
37	P0.25	Digital I/O	General purpose I/O <sup>2</sup>
38	P0.26	Digital I/O	General purpose I/O <sup>2</sup>
39	P0.27	Digital I/O	General purpose I/O <sup>2</sup>
40	P0.28	Digital I/O	General purpose I/O <sup>2</sup>
	AIN4	Analog input	SAADC/COMP/LPCOMP input
41	P0.29	Digital I/O	General purpose I/O <sup>2</sup>
	AIN5	Analog input	SAADC/COMP/LPCOMP input
42	P0.30	Digital I/O	General purpose I/O <sup>2</sup>
	AIN6	Analog input	SAADC/COMP/LPCOMP input
43	P0.31	Digital I/O	General purpose I/O pin <sup>2</sup>
	AIN7	Analog input	SAADC/COMP/LPCOMP input

Pin	Name	Type	Description
44	NC		No connect Leave unconnected
45	VSS	Power	Ground
46	DEC4	Power	1.3 V regulator supply decoupling Input from DC/DC regulator Output from 1.3 V LDO
47	DCC	Power	DC/DC regulator output
48	VDD	Power	Power supply
<b>Bottom of chip</b>			
Die pad	VSS	Power	Ground pad Exposed die pad must be connected to ground (VSS) for proper device operation.

## 4.2 WLCSP ball assignments

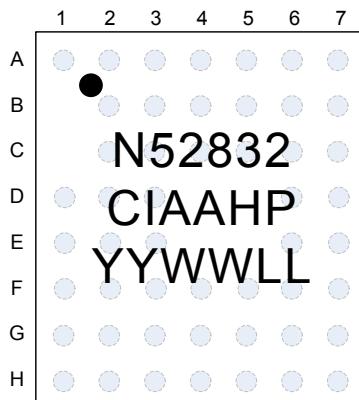


Figure 3: WLCSP ball assignments, top view

Table 3: WLCSP ball assignments

Ball	Name	Description	
A1	XC2	Analog input	Connection for 32 MHz crystal
A2	DEC2	Power	1.3 V regulator supply decoupling (Radio supply)
A3	P0.28	Digital I/O	General purpose I/O <sup>3</sup>
	AIN4	Analog input	SAADC/COMP/LPCOMP input
A4	P0.29	Digital I/O	General purpose I/O <sup>3</sup>
	AIN5	Analog input	SAADC/COMP/LPCOMP input
A5	P0.30	Digital I/O	General purpose I/O <sup>3</sup>
	AIN6	Analog input	SAADC/COMP/LPCOMP input
A6	DEC4	Power	1.3 V regulator supply decoupling Input from DC/DC converter. Output from 1.3 V LDO
A7	VDD	Power	Power supply
B2	XC1	Analog input	Connection for 32 MHz crystal
B3	P0.25	Digital I/O	General purpose I/O <sup>3</sup>

<sup>1</sup> See [GPIO located near the radio](#) on page 17 for more information.

<sup>2</sup> See [NFC antenna pins](#) on page 17 for more information.

Ball	Name	Description
B4	P0.27	Digital I/O General purpose I/O <sup>3</sup>
B5	P0.31	Digital I/O General purpose I/O <sup>3</sup>
	AIN7	Analog input SAADC/COMP/LPCOMP input
B6	DCC	Power DC/DC converter output
B7	DEC1	Power 0.9 V regulator digital supply decoupling
C2	DEC3	Power Power supply decoupling
C3	NC	N/A Not connected
C4	VSS	Power Ground
C5	VSS	Power Ground
C6	P0.02	Digital I/O General purpose I/O
	AIN0	Analog input SAADC/COMP/LPCOMP input
C7	P0.01	Digital I/O General purpose I/O
	XL2	Analog input Connection for 32.768 kHz crystal (LFXO)
D1	ANT	RF Single-ended radio antenna connection
D2	VSS_PA	Power Ground (Radio supply)
D3	P0.26	Digital I/O General purpose I/O <sup>3</sup>
D6	P0.03	Digital I/O General purpose I/O
	AIN1	Analog input SAADC/COMP/LPCOMP input
D7	P0.00	Digital I/O General purpose I/O
	XL1	Analog input Connection for 32.768 kHz crystal (LFXO)
E1	P0.24	Digital I/O General purpose I/O <sup>3</sup>
E2	P0.23	Digital I/O General purpose I/O <sup>3</sup>
E3	VSS	Power Ground
E6	P0.04	Digital I/O General purpose I/O
	AIN2	Analog input SAADC/COMP/LPCOMP input
E7	P0.05	Digital I/O General purpose I/O
	AIN3	Analog input SAADC/COMP/LPCOMP input
F1	SWDCLK	Digital input Serial wire debug clock input for debug and programming
F2	P0.22	Digital I/O General purpose I/O <sup>3</sup>
F3	P0.19	Digital I/O General purpose I/O
F4	P0.11	Digital I/O General purpose I/O
F5	VSS	Power Ground
F6	P0.07	Digital I/O General purpose I/O
F7	P0.06	Digital I/O General purpose I/O
G1	SWDIO	Digital I/O Serial wire debug I/O for debug and programming
G2	P0.20	Digital I/O General purpose I/O
	TRACECLK	Trace port clock output
G3	P0.17	Digital I/O General purpose I/O
G4	P0.13	Digital I/O General purpose I/O
G5	NFC2	NFC input NFC antenna connection
	P0.10	Digital I/O General purpose I/O <sup>4</sup>
G6	NFC1	NFC input NFC antenna connection
	P0.09	Digital I/O General purpose I/O <sup>4</sup>
G7	P0.08	Digital I/O General purpose I/O
H1	P0.21	Digital I/O General purpose I/O
	nRESET	Configurable as pin reset
H2	P0.18	Digital I/O General purpose I/O
	TRACEDATA[0]	Trace port output
H3	P0.16	Digital I/O General purpose I/O
	TRACEDATA[1]	Trace port output
H4	P0.15	Digital I/O General purpose I/O

Ball	Name	Description	
	TRACEDATA[2]		Trace port output
H5	P0.14	Digital I/O	General purpose I/O
	TRACEDATA[3]		Trace port output
H6	P0.12	Digital I/O	General purpose I/O
H7	VDD	Power	Power supply

## 4.3 GPIO usage restrictions

### 4.3.1 GPIO located near the radio

Radio performance parameters, such as sensitivity, may be affected by high frequency digital I/O with large sink/source current close to the Radio power supply and antenna pins.

*Table 4: GPIO recommended usage for QFN48 package* on page 17 and *Table 5: GPIO recommended usage for WLCSP package* on page 17 identify some GPIO that have recommended usage guidelines to maximize radio performance in an application.

**Table 4: GPIO recommended usage for QFN48 package**

Pin	GPIO	Recommended usage
27	P0.22	Low drive, low frequency I/O only.
28	P0.23	
29	P0.24	
37	P0.25	
38	P0.26	
39	P0.27	
40	P0.28	
41	P0.29	
42	P0.30	
43	P0.31	

**Table 5: GPIO recommended usage for WLCSP package**

Pin	GPIO	Recommended usage
F2	P0.22	Low drive, low frequency I/O only.
E2	P0.23	
E1	P0.24	
B3	P0.25	
D3	P0.26	
B4	P0.27	
A3	P0.28	
A4	P0.29	
A5	P0.30	
B5	P0.31	

### 4.3.2 NFC antenna pins

Two physical pins can be configured either as NFC antenna pins (factory default), or as GPIOs, as shown below.

**Table 6: GPIO pins used by NFC**

NFC pad name	GPIO
NFC1	P0.09
NFC2	P0.10

When configured as NFC antenna pins, the GPIOs on those pins will automatically be set to DISABLE state and a protection circuit will be enabled preventing the chip from being damaged in the presence of a strong NFC field. The protection circuit will short the two pins together if voltage difference exceeds approximately 2 V.

<sup>3</sup> See [GPIO located near the radio](#) on page 17 for more information.

<sup>4</sup> See [NFC antenna pins](#) on page 17 for more information.

For information on how to configure these pins as normal GPIOs, see [NFCT — Near field communication tag](#) on page 416 and [UICR — User information configuration registers](#) on page 54. Note that the device will not be protected against strong NFC field damage if the pins are configured as GPIO and an NFC antenna is connected to the device. The pins will always be configured as NFC pins during power-on reset until the configuration is set according to the UICR register.

These two pins will have some limitations when configured as GPIO. The pin capacitance will be higher on these pins, and there is some current leakage between the two pins if they are driven to different logical values. To avoid leakage between the pins when configured as GPIO, these GPIOs should always be at the same logical value whenever entering one of the device power saving modes. See [Electrical specification](#).

## 5 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.

**Table 7: Absolute maximum ratings**

	Min.	Max.	Unit
<b>Supply voltages</b>			
VDD	-0.3	+3.9	V
VSS		0	V
<b>I/O pin voltage</b>			
$V_{I/O}$ , $VDD \leq 3.6$ V	-0.3	$VDD + 0.3$ V	V
$V_{I/O}$ , $VDD > 3.6$ V	-0.3	3.9 V	V
<b>NFC antenna pin current</b>			
$I_{NFC1/2}$		80	mA
<b>Radio</b>			
RF input level		10	dBm
<b>Environmental QFN48, 6x6 mm package</b>			
Storage temperature	-40	+125	°C
MSL (moisture sensitivity level)		2	
ESD HBM (human body model)		4	kV
ESD CDM (charged device model)		1000	V
<b>Environmental WLCP, 3.0x3.2 mm package</b>			
Storage temperature	-40	+125	°C
MSL		1	
ESD HBM		2	kV
ESD CDM		500	V
<b>Flash memory</b>			
Endurance	10 000		Write/erase cycles
Retention	10 years at 40°C		



## 6 Recommended operating conditions

The operating conditions are the physical parameters that the chip can operate within.

**Table 8: Recommended operating conditions**

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

**Important:** The on-chip power-on reset circuitry may not function properly for rise times longer than the specified maximum.

# 7 CPU

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

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

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

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

Real-time execution is highly deterministic in thread mode, to and from sleep modes, and when handling events at configurable priority levels via the Nested Vectored Interrupt Controller (NVIC).

Executing code from flash will have a wait state penalty on the nRF52 Series. An instruction cache can be enabled to minimize flash wait states when fetching instructions. For more information on cache, see [Cache](#) on page 30. The section [Electrical specification](#) on page 21 shows CPU performance parameters including wait states in different modes, CPU current and efficiency, and processing power and efficiency based on the CoreMark® benchmark.

## 7.1 Floating point interrupt

The floating point unit (FPU) may generate exceptions when used due to e.g. overflow or underflow. These exceptions will trigger the FPU interrupt (see [Instantiation](#) on page 24). To clear the IRQ line when an exception has occurred, the relevant exception bit within the FPSCR register needs to be cleared. For more information about the FPSCR or other FPU registers, see [Cortex-M4 Devices Generic User Guide](#).

## 7.2 Electrical specification

### 7.2.1 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 <sub>FLASH</sub>	CPU wait states, running from flash, cache disabled	0	2		
W <sub>FLASHCACHE</sub>	CPU wait states, running from flash, cache enabled	0	3		
W <sub>RAM</sub>	CPU wait states, running from RAM		0		
I <sub>DDFLASHCACHE</sub>	CPU current, running from flash, cache enabled, LDO	7.4			mA
I <sub>DDFLASHCACHEDCDC</sub>	CPU current, running from flash, cache enabled, DCDC 3V	3.7			mA
I <sub>DDFLASH</sub>	CPU current, running from flash, cache disabled, LDO	8.0			mA
I <sub>DDFLASHCDC</sub>	CPU current, running from flash, cache disabled, DCDC 3V	3.9			mA
I <sub>DDRAM</sub>	CPU current, running from RAM, LDO	6.7			mA
I <sub>DDRAMCDC</sub>	CPU current, running from RAM, DCDC 3V	3.3			mA
I <sub>DDFLASH/MHz</sub>	CPU efficiency, running from flash, cache enabled, LDO	125			µA/ MHz
I <sub>DDFLASHCDC/MHz</sub>	CPU efficiency, running from flash, cache enabled, DCDC 3V	58			µA/ MHz

Symbol	Description	Min.	Typ.	Max.	Units
CM <sub>FLASH</sub>	CoreMark <sup>5</sup> , running from flash, cache enabled	215		CoreMark	
CM <sub>FLASH/MHz</sub>	CoreMark per MHz, running from flash, cache enabled	3.36		CoreMark	MHz
CM <sub>FLASH/mA</sub>	CoreMark per mA, running from flash, cache enabled, DCDC 3V	58		CoreMark	mA

## 7.3 CPU and support module configuration

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

Option / Module	Description	Implemented
<b>Core options</b>		
NVIC	Nested Vector Interrupt Controller	37 vectors
PRIORITIES	Priority bits	3
WIC	Wakeup Interrupt Controller	NO
Endianness	Memory system endianness	Little endian
Bit Banding	Bit banded memory	NO
DWT	Data Watchpoint and Trace	YES
SysTick	System tick timer	YES
<b>Modules</b>		
MPU	Memory protection unit	YES
FPU	Floating point unit	YES
DAP	Debug Access Port	YES
ETM	Embedded Trace Macrocell	YES
ITM	Instrumentation Trace Macrocell	YES
TPIU	Trace Port Interface Unit	YES
ETB	Embedded Trace Buffer	NO
FPB	Flash Patch and Breakpoint Unit	YES
HTM	AHB Trace Macrocell	NO

<sup>5</sup> Using IAR v6.50.1.4452 with flags --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4\_sp --Ohs --no\_size\_constraints

# 8 Memory

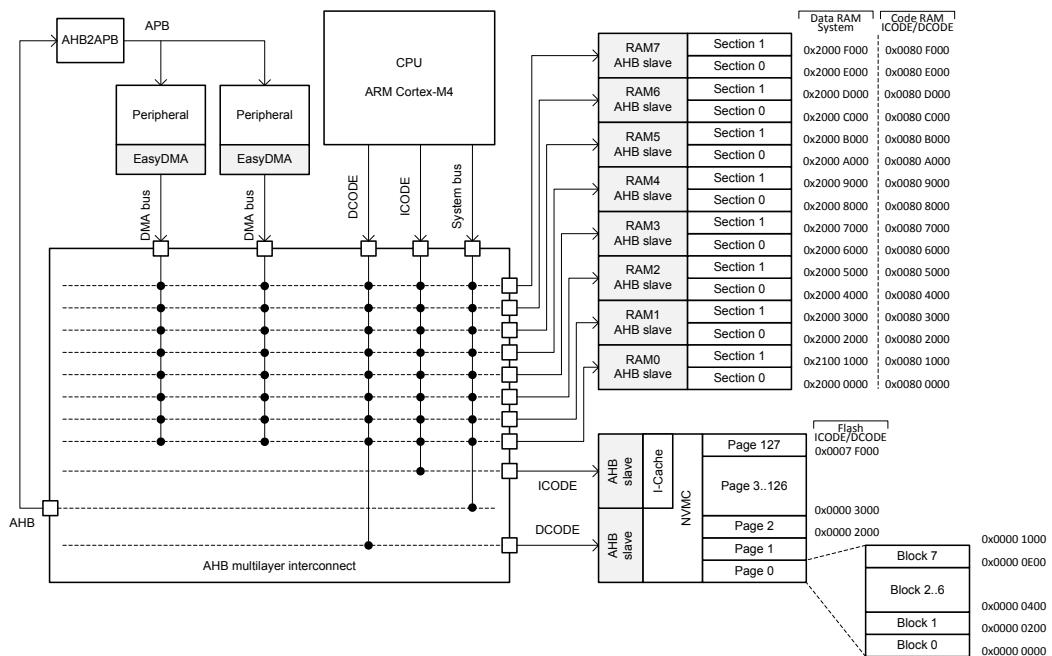
The nRF52832 contains flash and RAM that can be used for code and data storage.

The amount of RAM and flash will vary depending on variant, see [Table 9: Memory variants](#) on page 23.

**Table 9: Memory variants**

Device name	RAM	Flash	Comments
nRF52832-QFAA	64 kB	512 kB	
nRF52832-QFAB	32 kB	256 kB	
nRF52832-CIAA	64 kB	512 kB	

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 [Figure 4: Memory layout](#) on page 23.



**Figure 4: Memory layout**

See [AHB multilayer](#) on page 26 and [EasyDMA](#) on page 27 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.

## 8.1 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 [Figure 4: Memory layout](#) on page 23.

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 78).

## 8.2 Flash - Non-volatile memory

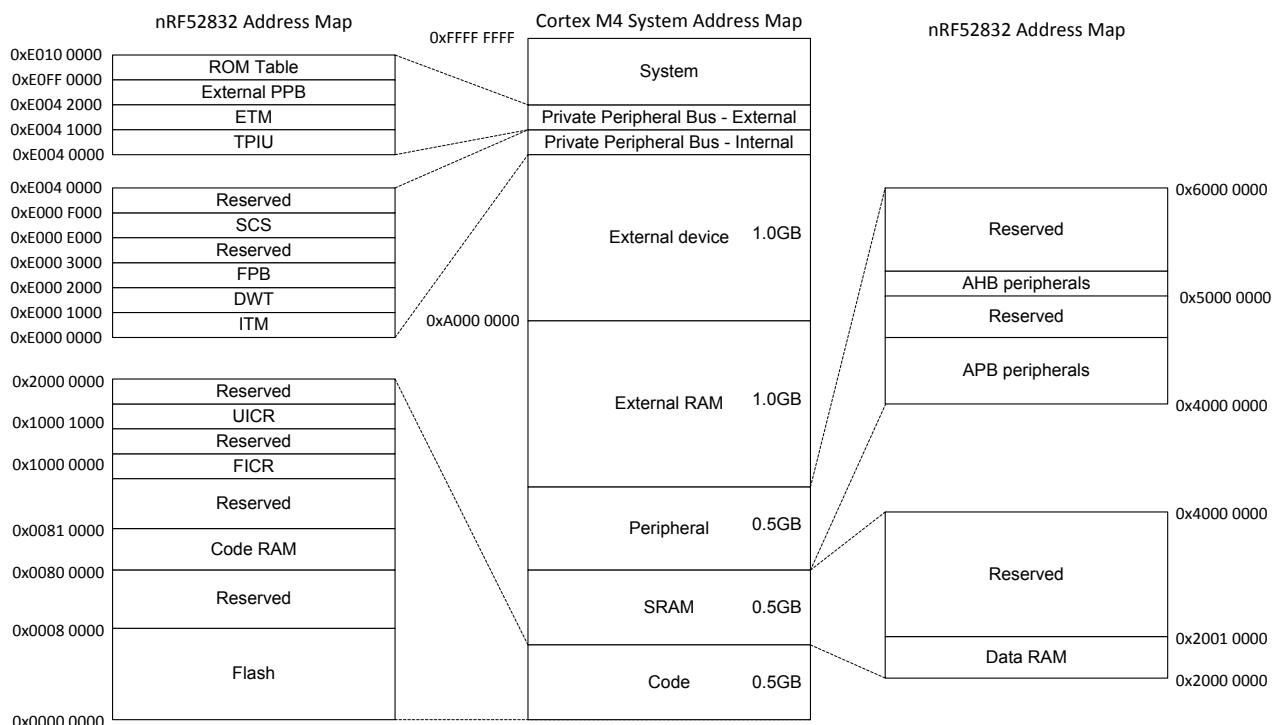
The Flash can be read an unlimited number of times by the CPU, but it has restrictions on the number of times it can be written and erased and also on how it can be written.

Writing to Flash is managed by the Non-volatile memory controller (NVMC), see [NVMC — Non-volatile memory controller](#) on page 29.

The Flash is divided into multiple pages that can be accessed by the CPU via both the ICODE and DCODE buses as shown in, [Figure 4: Memory layout](#) on page 23. Each page is divided into 8 blocks.

## 8.3 Memory map

The complete memory map is shown in [Figure 5: Memory map](#) on page 24. As described in [Memory](#) on page 23, Code RAM and the Data RAM are the same physical RAM.



**Figure 5: Memory map**

## 8.4 Instantiation

**Table 10: Instantiation table**

ID	Base Address	Peripheral	Instance	Description	
0	0x40000000	CLOCK	CLOCK	Clock control	
0	0x40000000	POWER	POWER	Power control	
0	0x40000000	BPROT	BPROT	Block Protect	
1	0x40001000	RADIO	RADIO	2.4 GHz radio	
2	0x40002000	UARTE	UARTE0	Universal Asynchronous Receiver/Transmitter with EasyDMA	
2	0x40002000	UART	UART0	Universal Asynchronous Receiver/Transmitter	Deprecated
3	0x40003000	SPIM	SPIMO	SPI master 0	
3	0x40003000	SPIS	SPISO	SPI slave 0	
3	0x40003000	TWIM	TWIMO	Two-wire interface master 0	
3	0x40003000	TWI	TWI0	Two-wire interface master 0	Deprecated
3	0x40003000	SPI	SPIO	SPI master 0	Deprecated

ID	Base Address	Peripheral	Instance	Description	
3	0x40003000	TWIS	TWISO	Two-wire interface slave 0	
4	0x40004000	SPIM	SPIM1	SPI master 1	
4	0x40004000	TWI	TWI1	Two-wire interface master 1	Deprecated
4	0x40004000	SPIS	SPIS1	SPI slave 1	
4	0x40004000	TWIS	TWIS1	Two-wire interface slave 1	
4	0x40004000	TWIM	TWIM1	Two-wire interface master 1	
4	0x40004000	SPI	SPI1	SPI master 1	Deprecated
5	0x40005000	NFCT	NFCT	Near Field Communication Tag	
6	0x40006000	GPIOTE	GPIOTE	GPIO Tasks and Events	
7	0x40007000	SAADC	SAADC	Analog to digital converter	
8	0x40008000	TIMER	TIMER0	Timer 0	
9	0x40009000	TIMER	TIMER1	Timer 1	
10	0x4000A000	TIMER	TIMER2	Timer 2	
11	0x4000B000	RTC	RTC0	Real-time counter 0	
12	0x4000C000	TEMP	TEMP	Temperature sensor	
13	0x4000D000	RNG	RNG	Random number generator	
14	0x4000E000	ECB	ECB	AES Electronic Code Book (ECB) mode block encryption	
15	0x4000F000	CCM	CCM	AES CCM Mode Encryption	
15	0x4000F000	AAR	AAR	Acelerated Address Resolver	
16	0x40010000	WDT	WDT	Watchdog timer	
17	0x40011000	RTC	RTC1	Real-time counter 1	
18	0x40012000	QDEC	QDEC	Quadrature decoder	
19	0x40013000	LPCOMP	LPCOMP	Low power comparator	
19	0x40013000	COMP	COMP	General purpose comparator	
20	0x40014000	SWI	SWI0	Software interrupt 0	
20	0x40014000	EGU	EGU0	Event Generator Unit 0	
21	0x40015000	EGU	EGU1	Event Generator Unit 1	
21	0x40015000	SWI	SWI1	Software interrupt 1	
22	0x40016000	SWI	SWI2	Software interrupt 2	
22	0x40016000	EGU	EGU2	Event Generator Unit 2	
23	0x40017000	SWI	SWI3	Software interrupt 3	
23	0x40017000	EGU	EGU3	Event Generator Unit 3	
24	0x40018000	EGU	EGU4	Event Generator Unit 4	
24	0x40018000	SWI	SWI4	Software interrupt 4	
25	0x40019000	SWI	SWI5	Software interrupt 5	
25	0x40019000	EGU	EGU5	Event Generator Unit 5	
26	0x4001A000	TIMER	TIMER3	Timer 3	
27	0x4001B000	TIMER	TIMER4	Timer 4	
28	0x4001C000	PWM	PWMO	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	
32	0x40020000	MWU	MWU	Memory Watch Unit	
33	0x40021000	PWM	PWM1	Pulse Width Modulation Unit 1	
34	0x40022000	PWM	PWM2	Pulse Width Modulation Unit 2	
35	0x40023000	SPI	SPI2	SPI master 2	Deprecated
35	0x40023000	SPIS	SPIS2	SPI slave 2	
35	0x40023000	SPIM	SPIM2	SPI master 2	
36	0x40024000	RTC	RTC2	Real-time counter 2	
37	0x40025000	I2S	I2S	Inter-IC Sound Interface	
38	0x40026000	FPU	FPU	FPU interrupt	
0	0x50000000	GPIO	GPIO	General purpose input and output	Deprecated
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	

## 9 AHB multilayer

The CPU and all of the EasyDMAs are AHB bus masters on the AHB multilayer, while the RAM and various other modules are AHB slaves.

See [Block diagram](#) on page 12 for an overview of which peripherals implement EasyDMA.

The CPU has exclusive access to all AHB slaves except for the RAM that can also be accessed by the EasyDMA.

Access rights to each of the RAM AHB slaves are resolved using the priority of the different bus masters in the system

See [AHB multilayer priorities](#) on page 26 for information about the priority of the different AHB bus masters in the system. It is possible for two or more bus masters to have the same priority in cases where it is guaranteed by design that the related masters will never be able to access the same slave at the same time.

### 9.1 AHB multilayer priorities

Each master connected to the AHB multilayer is assigned a priority.

**Table 11: AHB bus masters**

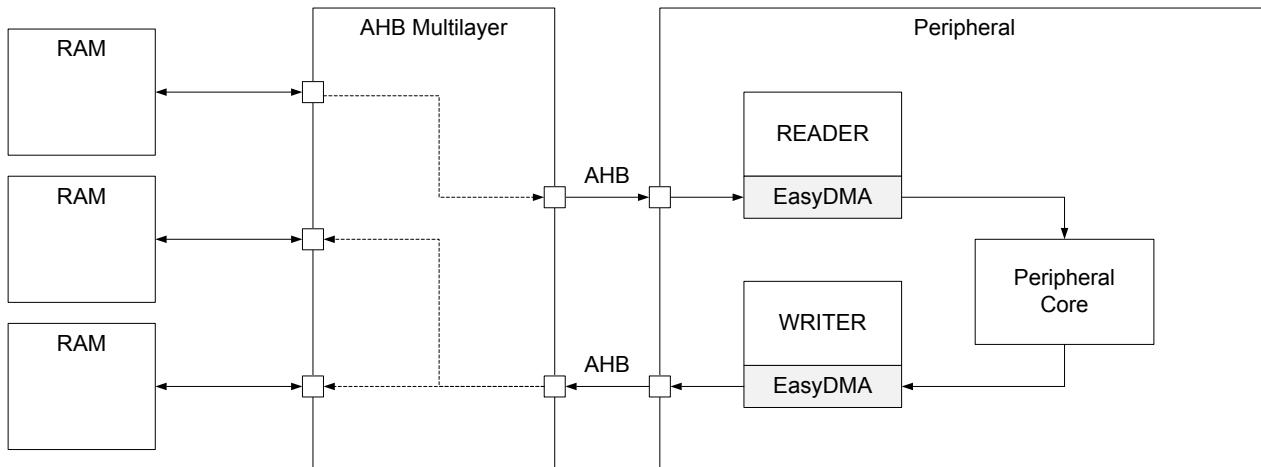
Bus master name	Priority	Description
CPU	Highest priority	
SPIS1		Applies to SPIM1, SPIS1, TWIM1, TWIS1
RADIO		
CCM/ECB/AAR		
SAADC		
UARTE		
SERIAL0		Applies to SPIM0, SPIS0, TWIM0, TWIS0
SERIAL2		Applies to SPIM2, SPIS2
NFC		
I2S	I2S	
PDM	PDM	
PWM	Lowest priority	Applies to PWM0, PWM1, PWM2

# 10 EasyDMA

EasyDMA is an easy-to-use direct memory access module that some peripherals implement to gain direct access to Data RAM.

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

A peripheral can implement multiple EasyDMA instances, for example to provide a dedicated channel for reading data from RAM into the peripheral at the same time as a second channel is dedicated for writing data to the RAM from the peripheral. This concept is illustrated in [Figure 6: EasyDMA example](#) on page 27



**Figure 6: EasyDMA example**

An EasyDMA channel is usually exposed to the user in the form illustrated 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 here assumed that the peripheral will read 5 bytes from the readerBuffer located in RAM at address 0x20000000, process the data and then 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 [Figure 7: EasyDMA memory layout](#) on page 28.

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 7: EasyDMA memory layout**

The EasyDMA channel's MAXCNT register cannot be specified larger than the actual size of the buffer. If, for example, the WRITER.MAXCNT register is specified larger than the size of the writerBuffer, the WRITER EasyDMA channel may overflow the writerBuffer.

After the peripheral has completed the EasyDMA transfer, the CPU can read the EasyDMA channel's AMOUNT register to see how many bytes that were transferred, e.g. it is possible for the CPU to read the MYPERIPHERAL->WRITER.AMOUNT register to see how many bytes the WRITER wrote to RAM.

## 10.1 EasyDMA array list

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

The EasyDMA array list can be represented by the data structure ArrayList\_type illustrated in the code example below.

This data structure includes only a buffer with size equal to READER.MAXCNT. EasyDMA will use the READER.MAXCNT register to determine when the buffer is full.

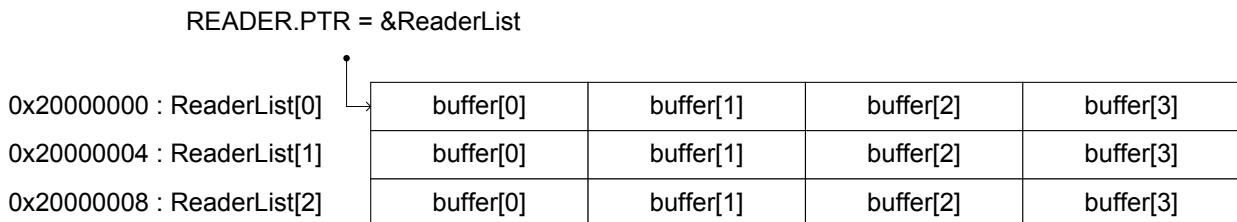
This array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

```
#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;
```



**Figure 8: EasyDMA array list**

# 11 NVMC — Non-volatile memory controller

The Non-volatile memory controller (NVMC) is used for writing and erasing the internal Flash memory and the UICR.

Before a write can be performed, the NVMC must be enabled for writing in CONFIG.WEN. Similarly, before an erase can be performed, the NVMC must be enabled for erasing in CONFIG.EEN, see [CONFIG](#) on page 31. The user must make sure that writing and erasing are not enabled at the same time. Failing to do so may result in unpredictable behavior.

## 11.1 Writing to Flash

When writing is enabled, the Flash is written by writing a full 32-bit word to a word-aligned address in the Flash.

The NVMC is only able to write '0' to bits in the Flash that are erased, that is, set to '1'. It cannot write back a bit to '1'.

As illustrated in [Memory](#) on page 23, the Flash is divided into multiple pages that are further divided into multiple blocks. The same block in the 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 23 for block size.

Only full 32-bit words can be written to Flash using the NVMC interface. To write less than 32 bits to Flash, write the data as a word, and set all the bits that should remain unchanged in the word to '1'. Note that the restriction about the number of writes (see above) still applies in this case.

The time it takes to write a word to the Flash is specified by  $t_{WRITE}$ . The CPU is halted while the NVMC is writing to the Flash.

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

## 11.2 Erasing a page in Flash

When erase is enabled, the Flash 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 while the NVMC performs the erase operation.

## 11.3 Writing to user information configuration registers (UICR)

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

UICR can only be written  $n_{WRITE}$  number of times before an erase must be performed using [ERASEUICR](#) or [ERASEALL](#).

The time it takes to write a word to the UICR is specified by  $t_{WRITE}$ . The CPU is halted while the NVMC is writing to the UICR.

## 11.4 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 while the NVMC performs the erase operation.

## 11.5 Erase all

When erase is enabled, the whole Flash and UICR can be erased 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 [tERASEALL](#). The CPU is halted while the NVMC performs the erase operation.

## 11.6 Cache

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

See the Memory map in [Memory map](#) on page 24 for the location of Flash.

A cache hit is an instruction fetch from the cache, and it has a 0 wait-state delay. The number of wait-states for a cache miss, where the instruction is not available in the cache and needs to be fetched from Flash, depends on the processor frequency and is shown in [CPU](#) on page 21.

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

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

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

## 11.7 Registers

**Table 12: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4001E000	NVMC	NVMC	Non-Volatile Memory Controller	

**Table 13: Register Overview**

Register	Offset	Description	
<a href="#">READY</a>	0x400	Ready flag	
<a href="#">CONFIG</a>	0x504	Configuration register	
<a href="#">ERASEPAGE</a>	0x508	Register for erasing a page in Code area	
<a href="#">ERASEPCR1</a>	0x508	Register for erasing a page in Code area. Equivalent to ERASEPAGE.	Deprecated
<a href="#">ERASEALL</a>	0x50C	Register for erasing all non-volatile user memory	
<a href="#">ERASEPCRO</a>	0x510	Register for erasing a page in Code area. Equivalent to ERASEPAGE.	Deprecated
<a href="#">ERASEUICR</a>	0x514	Register for erasing User Information Configuration Registers	
<a href="#">ICACHECNF</a>	0x540	I-Code cache configuration register.	
<a href="#">IHIT</a>	0x548	I-Code cache hit counter.	
<a href="#">IMISS</a>	0x54C	I-Code cache miss counter.	

### 11.7.1 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 (on-going write or erase operation)																										
		Ready		1		NVMC is ready																										

## 11.7.2 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																														
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	WEN			Program memory access mode. It is strongly recommended to only activate erase and write modes when they are actively used. Enabling write or erase will invalidate the cache and keep it invalidated.																											
		Ren	0		Read only access																											
		Wen	1		Write Enabled																											
		Een	2		Erase enabled																											

### 11.7.3 ERASEPAGE

Address offset: 0x508

Register for erasing a page in Code area

## 11.7.4 ERASEPCR1 ( Deprecated )

Address offset: 0x508

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

## **11.7.5 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																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 code erase has to be enabled by CONFIG.EEN before the UICR can be erased.																											
		NoOperation	0		No operation																											
		Erase	1		Start chip erase																											

### **11.7.6 ERASEPCR0 ( Deprecated )**

Address offset: 0x510

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

## 11.7.7 ERASEUICR

Address offset: 0x514

## Register for erasing User Information Configuration Registers

### **11.7.8 ICACHECNF**

Address offset: 0x540

## I-Code cache configuration register.

## 11.7.9 IHIT

Address offset: 0x548

## I-Code cache hit counter.

## **11.7.10 IMISS**

Address offset: 0x54C

## I-Code cache miss counter.

## **11.8 Electrical specification**

### 11.8.1 Flash programming

Symbol	Description	Min.	Typ.	Max.	Units
$n_{\text{WRITE,BLOCK}}$	Amount of writes allowed in a block between erase			181	
$t_{\text{WRITE}}$	Time to write one word		67.5	338	$\mu\text{s}$
$t_{\text{ERASEPAGE}}$	Time to erase one page		2.05	89.7	ms
$t_{\text{ERASEALL}}$	Time to erase all flash		6.72	295.3	ms

## 11.8.2 Cache size

Symbol	Description	Min.	Typ.	Max.	Units
Size <sub>I<sup>CODE</sup></sub>	I-Code cache size		2048		Bytes

# 12 BPROT — Block protection

The mechanism for protecting non-volatile memory can be used to prevent 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 four CONFIG registers of 32 bits, which means there are 128 protectable blocks in total.

**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 interface mode (a debugger is connected) and the DISABLEINDEBUG register is set to disable. For more information, see [Debug and trace](#) on page 72.

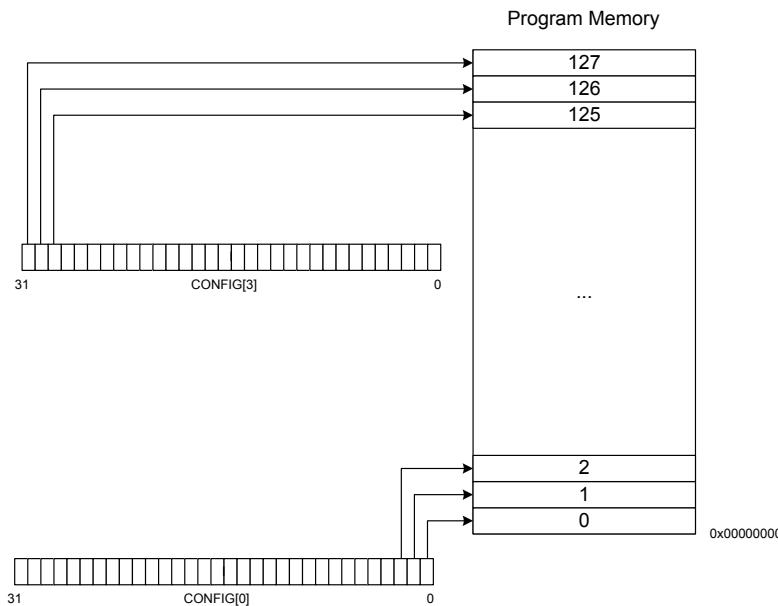


Figure 9: Protected regions of program memory

## 12.1 Registers

Table 14: Instances

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

Table 15: Register Overview

Register	Offset	Description
<a href="#">CONFIG0</a>	0x600	Block protect configuration register 0
<a href="#">CONFIG1</a>	0x604	Block protect configuration register 1
<a href="#">DISABLEINDEBUG</a>	0x608	Disable protection mechanism in debug interface mode
	0x60C	Reserved
<a href="#">CONFIG2</a>	0x610	Block protect configuration register 2
<a href="#">CONFIG3</a>	0x614	Block protect configuration register 3

## 12.1.1 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	
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 enable																												
B	RW	REGION1				Enable protection for region 1. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
C	RW	REGION2				Enable protection for region 2. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
D	RW	REGION3				Enable protection for region 3. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
E	RW	REGION4				Enable protection for region 4. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
F	RW	REGIONS5				Enable protection for region 5. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
G	RW	REGION6				Enable protection for region 6. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
H	RW	REGION7				Enable protection for region 7. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
I	RW	REGION8				Enable protection for region 8. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
J	RW	REGION9				Enable protection for region 9. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
K	RW	REGION10				Enable protection for region 10. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
L	RW	REGION11				Enable protection for region 11. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
M	RW	REGION12				Enable protection for region 12. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
N	RW	REGION13				Enable protection for region 13. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
O	RW	REGION14				Enable protection for region 14. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
P	RW	REGION15				Enable protection for region 15. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												
		Enabled	1			Protection enable																												
Q	RW	REGION16				Enable protection for region 16. Write '0' has no effect.																												
		Disabled	0			Protection disabled																												

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			f	e	d	c	b	a	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		
<b>Id</b>	<b>RW</b>		<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																											
							Enabled	1		Protection enable																								
R	RW	REGION17	Disabled	0			Enable protection for region 17. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
S	RW	REGION18	Disabled	0			Enable protection for region 18. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
T	RW	REGION19	Disabled	0			Enable protection for region 19. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
U	RW	REGION20	Disabled	0			Enable protection for region 20. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
V	RW	REGION21	Disabled	0			Enable protection for region 21. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
W	RW	REGION22	Disabled	0			Enable protection for region 22. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
X	RW	REGION23	Disabled	0			Enable protection for region 23. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
Y	RW	REGION24	Disabled	0			Enable protection for region 24. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
Z	RW	REGION25	Disabled	0			Enable protection for region 25. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
a	RW	REGION26	Disabled	0			Enable protection for region 26. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
b	RW	REGION27	Disabled	0			Enable protection for region 27. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
c	RW	REGION28	Disabled	0			Enable protection for region 28. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
d	RW	REGION29	Disabled	0			Enable protection for region 29. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
e	RW	REGION30	Disabled	0			Enable protection for region 30. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											
f	RW	REGION31	Disabled	0			Enable protection for region 31. Write '0' has no effect.																											
			Enabled	1			Protection disabled																											
							Protection enable																											

## 12.1.2 CONFIG1

Address offset: 0x604

Block protect configuration register 1

				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
				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	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>		<b>Value</b>		<b>Description</b>																												
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																												
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																												
Q	RW	REGION48					Enable protection for region 48. Write '0' has no effect.																												
			Disabled		0		Protection disabled																												
			Enabled		1		Protection enabled																												
R	RW	REGION49					Enable protection for region 49. Write '0' has no effect.																												
			Disabled		0		Protection disabled																												
			Enabled		1		Protection enabled																												
S	RW	REGION50					Enable protection for region 50. Write '0' has no effect.																												

			31	30	29	28	27	26	25	24	23	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	
Id			Value Id																															
T			RW																															
T			REGION51																															
T			Disabled																															
T			Enabled																															
U			RW																															
U			REGION52																															
U			Disabled																															
U			Enabled																															
V			RW																															
V			REGION53																															
V			Disabled																															
V			Enabled																															
W			RW																															
W			REGION54																															
W			Disabled																															
W			Enabled																															
X			RW																															
X			REGION55																															
X			Disabled																															
X			Enabled																															
Y			RW																															
Y			REGION56																															
Y			Disabled																															
Y			Enabled																															
Z			RW																															
Z			REGION57																															
Z			Disabled																															
Z			Enabled																															
a			RW																															
a			REGION58																															
a			Disabled																															
a			Enabled																															
b			RW																															
b			REGION59																															
b			Disabled																															
b			Enabled																															
c			RW																															
c																																		

### 12.1.3 DISABLEINDEBUG

Address offset: 0x608

Disable protection mechanism in debug interface 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																															
<b>Reset 0x00000001</b>	0	0	0	0	0	0	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 interface mode. This register will only disable the protection mechanism if the device is in debug interface mode.																											
		Disabled	1	Disable in debug																												

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id	A		
<b>Reset 0x00000001</b>	0 1		
<b>Id    RW    Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
	Enabled	0	Enable in debug

## 12.1.4 CONFIG2

Address offset: 0x610

## Block protect configuration register 2



## 12.1.5 CONFIG3

Address offset: 0x614

Block protect configuration register 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		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		
Id	RW	Field	Value	Id	Value	Description																											
A	RW	REGION96				Enable protection for region 96. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
B	RW	REGION97				Enable protection for region 97. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
C	RW	REGION98				Enable protection for region 98. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
D	RW	REGION99				Enable protection for region 99. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
E	RW	REGION100				Enable protection for region 100. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
F	RW	REGION101				Enable protection for region 101. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
G	RW	REGION102				Enable protection for region 102. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
H	RW	REGION103				Enable protection for region 103. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
I	RW	REGION104				Enable protection for region 104. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
J	RW	REGION105				Enable protection for region 105. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
K	RW	REGION106				Enable protection for region 106. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
L	RW	REGION107				Enable protection for region 107. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
M	RW	REGION108				Enable protection for region 108. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
N	RW	REGION109				Enable protection for region 109. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
O	RW	REGION110				Enable protection for region 110. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
P	RW	REGION111				Enable protection for region 111. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											
			Enabled	1		Protection enabled																											
Q	RW	REGION112				Enable protection for region 112. Write '0' has no effect.																											
			Disabled	0		Protection disabled																											

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
			Enabled	1		Protection enabled																										
R	RW	REGION113	Disabled	0		Enable protection for region 113. Write '0' has no effect.																										
			Enabled	1		Protection disabled																										
S	RW	REGION114	Disabled	0		Protection enabled																										
			Enabled	1		Enable protection for region 114. Write '0' has no effect.																										
T	RW	REGION115	Disabled	0		Protection disabled																										
			Enabled	1		Protection enabled																										
U	RW	REGION116	Disabled	0		Enable protection for region 116. Write '0' has no effect.																										
			Enabled	1		Protection enabled																										
V	RW	REGION117	Disabled	0		Enable protection for region 117. Write '0' has no effect.																										
			Enabled	1		Protection disabled																										
W	RW	REGION118	Disabled	0		Protection enabled																										
			Enabled	1		Enable protection for region 118. Write '0' has no effect.																										
X	RW	REGION119	Disabled	0		Protection disabled																										
			Enabled	1		Protection enabled																										
Y	RW	REGION120	Disabled	0		Enable protection for region 120. Write '0' has no effect.																										
			Enabled	1		Protection enabled																										
Z	RW	REGION121	Disabled	0		Enable protection for region 121. Write '0' has no effect.																										
			Enabled	1		Protection disabled																										
a	RW	REGION122	Disabled	0		Protection enabled																										
			Enabled	1		Enable protection for region 122. Write '0' has no effect.																										
b	RW	REGION123	Disabled	0		Protection disabled																										
			Enabled	1		Enable protection for region 123. Write '0' has no effect.																										
c	RW	REGION124	Disabled	0		Protection enabled																										
			Enabled	1		Enable protection for region 124. Write '0' has no effect.																										
d	RW	REGION125	Disabled	0		Protection disabled																										
			Enabled	1		Enable protection for region 125. Write '0' has no effect.																										
e	RW	REGION126	Disabled	0		Protection enabled																										
			Enabled	1		Enable protection for region 126. Write '0' has no effect.																										
f	RW	REGION127	Disabled	0		Protection disabled																										
			Enabled	1		Enable protection for region 127. Write '0' has no effect.																										

# 13 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.

## 13.1 Registers

**Table 16: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x10000000	FICR	FICR	Factory Information Configuration	

**Table 17: Register Overview**

Register	Offset	Description
<i>CODEPAGESIZE</i>	0x010	Code memory page size
<i>CODESIZE</i>	0x014	Code memory size
<i>DEVICEID[0]</i>	0x060	Device identifier
<i>DEVICEID[1]</i>	0x064	Device identifier
<i>ER[0]</i>	0x080	Encryption Root, word 0
<i>ER[1]</i>	0x084	Encryption Root, word 1
<i>ER[2]</i>	0x088	Encryption Root, word 2
<i>ER[3]</i>	0x08C	Encryption Root, word 3
<i>IR[0]</i>	0x090	Identity Root, word 0
<i>IR[1]</i>	0x094	Identity Root, word 1
<i>IR[2]</i>	0x098	Identity Root, word 2
<i>IR[3]</i>	0x09C	Identity Root, word 3
<i>DEVICEADDRTYPE</i>	0x0A0	Device address type
<i>DEVICEADDR[0]</i>	0x0A4	Device address 0
<i>DEVICEADDR[1]</i>	0x0A8	Device address 1
<i>INFO.PART</i>	0x100	Part code
<i>INFO.VARIANT</i>	0x104	Part Variant, Hardware version and Production configuration
<i>INFO.PACKAGE</i>	0x108	Package option
<i>INFO.RAM</i>	0x10C	RAM variant
<i>INFO.FLASH</i>	0x110	Flash variant
	0x114	Reserved
	0x118	Reserved
	0x11C	Reserved
<i>TEMP.A0</i>	0x404	Slope definition A0.
<i>TEMP.A1</i>	0x408	Slope definition A1.
<i>TEMP.A2</i>	0x40C	Slope definition A2.
<i>TEMP.A3</i>	0x410	Slope definition A3.
<i>TEMP.A4</i>	0x414	Slope definition A4.
<i>TEMP.A5</i>	0x418	Slope definition A5.
<i>TEMP.B0</i>	0x41C	y-intercept B0.
<i>TEMP.B1</i>	0x420	y-intercept B1.
<i>TEMP.B2</i>	0x424	y-intercept B2.
<i>TEMP.B3</i>	0x428	y-intercept B3.
<i>TEMP.B4</i>	0x42C	y-intercept B4.
<i>TEMP.B5</i>	0x430	y-intercept B5.
<i>TEMP.T0</i>	0x434	Segment end T0.
<i>TEMP.T1</i>	0x438	Segment end T1.
<i>TEMP.T2</i>	0x43C	Segment end T2.
<i>TEMP.T3</i>	0x440	Segment end T3.
<i>TEMP.T4</i>	0x444	Segment end T4.

Register	Offset	Description
NFC.TAGHEADER0	0x450	Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER1	0x454	Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER2	0x458	Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER3	0x45C	Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.

### **13.1.1 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
Id	RW	Field	Value Id	Value	Description																											
A	R	CODEPAGESIZE		Code memory page size																												

### **13.1.2 CODESIZE**

Address offset: 0x014

## Code memory size

### 13.1.3 DEVICEID[0]

Address offset: 0x060

## Device identifier

### **13.1.4 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		
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	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															

### 13.1.5 ER[0]

Address offset: 0x080

## Encryption Root, word 0

### 13.1.6 ER[1]

Address offset: 0x084

## Encryption Root, word 1

### 13.1.7 ER[2]

Address offset: 0x088

## Encryption Root, word 2

### 13.1.8 ER[3]

Address offset: 0x08C

### Encryption Root, word 3

### 13.1.9 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		
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	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																										

### **13.1.10 IR[1]**

Address offset: 0x094

## Identity Root, word 1

### 13.1.11 IR[2]

Address offset: 0x098

## Identity Root, word 2

### 13.1.12 IR[3]

Address offset: 0x09C

## Identity Root, word 3

### **13.1.13 DEVICEADDRTYPE**

Address offset: 0x0A0

### Device address type

### 13.1.14 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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.																											

### **13.1.15 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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.																											

### **13.1.16 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
<b>Reset 0x00052832</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>		
Id	RW	Field	Value Id	Value	Description																											
A	R	PART			Part code																											
		N52832	0x52832	nRF52832																												
		Unspecified	0xFFFFFFFF	Unspecified																												

## **13.1.17 INFO.VARIANT**

Address offset: 0x104

## Part Variant, Hardware version and Production configuration

## **13.1.18 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	
<b>Reset 0x00002000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>												
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	R	PACKAGE			Package option																											
		QF	0x2000	QFx - 48-pin QFN																												
		CI	0x2001	Clx - 7x8 WLCSP 56 balls																												
		Unspecified	0xFFFFFFFF	Unspecified																												

### **13.1.19 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	A	
<b>Reset 0x00000040</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>		
Id	RW	Field	Value Id	Value	Description																											
A	R	RAM			RAM variant																											
		K16	0x10	16 kByte RAM																												
		K32	0x20	32 kByte RAM																												
		K64	0x40	64 kByte RAM																												
		Unspecified	0xFFFFFFFF	Unspecified																												

## **13.1.20 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	A	
<b>Reset 0x00000200</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>									
Id	RW	Field	Value Id	Value	Description																											
A	R	FLASH			Flash variant																											
		K128	0x80	128 kByte FLASH																												
		K256	0x100	256 kByte FLASH																												
		K512	0x200	512 kByte FLASH																												
		Unspecified	0xFFFFFFFF	Unspecified																												

### **13.1.21 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00000320</b>	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	
Id	RW	Field	Value Id	Value	Description																											
A	R	A		A (slope definition) register.																												

## 13.1.22 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000343</b>	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	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	R	A			A (slope definition) register.																											

13.1.23 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
<b>Reset 0x0000035D</b>	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	1	1	1	1	0	1	
Id	RW	Field	Value Id	Value	Description																												
A	R	A			A (slope definition) register.																												

13.1.24 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
<b>Reset 0x00000400</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																												
A	R	A			A (slope definition) register.																												

### **13.1.25 TEMP.A4**

Address offset: 0x414

## Slope definition A4.

### **13.1.26 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
<b>Reset 0x0000037B</b>	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	1	1	1	1	0	1	1
Id	RW	Field	Value Id	Value	Description																												
A	R	A			A (slope definition) register.																												

## 13.1.27 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00003FCC</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	B			B (y-intercept)																											

13.1.28 TEMP.B1

Address offset: 0x420

y-intercept B1.

13.1.29 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00003F98</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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)																											

### **13.1.30 TEMP.B3**

Address offset: 0x428

y-intercept B3.

### 13.1.31 TEMP.B4

Address offset: 0x42C

y-intercept B4.

13.1.32 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	
<b>Reset 0x00003E10</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	B			B (y-intercept)																											

### **13.1.33 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				
<b>Reset 0x000000E2</b>	0 1 1 1 0 0 0 1 0				
Id	RW	Field	Value Id	Value	Description
A	R	T			T (segment end)register.

### **13.1.34 TEMP.T1**

Address offset: 0x438

Segment end T1.

### **13.1.35 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				
<b>Reset 0x00000014</b>	<b>0 1 0 1 0 0</b>				
Id	RW	Field	Value Id	Value	Description
A	R	T			T (segment end)register.

### **13.1.36 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				
<b>Reset 0x00000019</b>	<b>0 1 1 0 0 1</b>				
Id	RW	Field	Value Id	Value	Description
A	R	T			T (segment end)register.

### **13.1.37 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				
<b>Reset 0x00000050</b>	0 1 0 1 0 0 0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	R	T			T (segment end)register.

### 13.1.38 NFC.TAGHEADER0

Address offset: 0x450

Default header for NFC Tag. Software can read these values to populate NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST.

### **13.1.39 NFC.TAGHEADER1**

Address offset: 0x454

Default header for NFC Tag. Software can read these values to populate NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST.

### **13.1.40 NFC.TAGHEADER2**

Address offset: 0x458

Default header for NFC Tag. Software can read these values to populate NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	D	D	D	D	D	D	C	C	C	C	C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	A	A	A	A		
Reset OxFFFFFFF	1	1	1	1	1	1	1	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	UD8			Unique identifier byte 8																											
B	R	UD9			Unique identifier byte 9																											
C	R	UD10			Unique identifier byte 10																											
D	R	UD11			Unique identifier byte 11																											

### **13.1.41 NFC.TAGHEADER3**

Address offset: 0x45C

Default header for NFC Tag. Software can read these values to populate NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	D	D	D	D	D	D	C	C	C	C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A		
<b>Reset 0xFFFFFFFF</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>			
Id	RW	Field	Value	Id	Description																											
A	R	UD12			Unique identifier byte 12																											
B	R	UD13			Unique identifier byte 13																											
C	R	UD14			Unique identifier byte 14																											
D	R	UD15			Unique identifier byte 15																											

# 14 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 29 and [Memory](#) on page 23 chapters.

## 14.1 Registers

**Table 18: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x10001000	UICR	UICR	User Information Configuration	

**Table 19: Register Overview**

Register	Offset	Description
	0x000	Reserved
	0x004	Reserved
	0x008	Reserved
	0x010	Reserved
<a href="#">NRFFW[0]</a>	0x014	Reserved for Nordic firmware design
<a href="#">NRFFW[1]</a>	0x018	Reserved for Nordic firmware design
<a href="#">NRFFW[2]</a>	0x01C	Reserved for Nordic firmware design
<a href="#">NRFFW[3]</a>	0x020	Reserved for Nordic firmware design
<a href="#">NRFFW[4]</a>	0x024	Reserved for Nordic firmware design
<a href="#">NRFFW[5]</a>	0x028	Reserved for Nordic firmware design
<a href="#">NRFFW[6]</a>	0x02C	Reserved for Nordic firmware design
<a href="#">NRFFW[7]</a>	0x030	Reserved for Nordic firmware design
<a href="#">NRFFW[8]</a>	0x034	Reserved for Nordic firmware design
<a href="#">NRFFW[9]</a>	0x038	Reserved for Nordic firmware design
<a href="#">NRFFW[10]</a>	0x03C	Reserved for Nordic firmware design
<a href="#">NRFFW[11]</a>	0x040	Reserved for Nordic firmware design
<a href="#">NRFFW[12]</a>	0x044	Reserved for Nordic firmware design
<a href="#">NRFFW[13]</a>	0x048	Reserved for Nordic firmware design
<a href="#">NRFFW[14]</a>	0x04C	Reserved for Nordic firmware design
<a href="#">NRFHW[0]</a>	0x050	Reserved for Nordic hardware design
<a href="#">NRFHW[1]</a>	0x054	Reserved for Nordic hardware design
<a href="#">NRFHW[2]</a>	0x058	Reserved for Nordic hardware design
<a href="#">NRFHW[3]</a>	0x05C	Reserved for Nordic hardware design
<a href="#">NRFHW[4]</a>	0x060	Reserved for Nordic hardware design
<a href="#">NRFHW[5]</a>	0x064	Reserved for Nordic hardware design
<a href="#">NRFHW[6]</a>	0x068	Reserved for Nordic hardware design
<a href="#">NRFHW[7]</a>	0x06C	Reserved for Nordic hardware design
<a href="#">NRFHW[8]</a>	0x070	Reserved for Nordic hardware design
<a href="#">NRFHW[9]</a>	0x074	Reserved for Nordic hardware design
<a href="#">NRFHW[10]</a>	0x078	Reserved for Nordic hardware design
<a href="#">NRFHW[11]</a>	0x07C	Reserved for Nordic hardware design
<a href="#">CUSTOMER[0]</a>	0x080	Reserved for customer
<a href="#">CUSTOMER[1]</a>	0x084	Reserved for customer
<a href="#">CUSTOMER[2]</a>	0x088	Reserved for customer
<a href="#">CUSTOMER[3]</a>	0x08C	Reserved for customer
<a href="#">CUSTOMER[4]</a>	0x090	Reserved for customer
<a href="#">CUSTOMER[5]</a>	0x094	Reserved for customer
<a href="#">CUSTOMER[6]</a>	0x098	Reserved for customer

Register	Offset	Description
<i>CUSTOMER[7]</i>	0x09C	Reserved for customer
<i>CUSTOMER[8]</i>	0x0A0	Reserved for customer
<i>CUSTOMER[9]</i>	0x0A4	Reserved for customer
<i>CUSTOMER[10]</i>	0x0A8	Reserved for customer
<i>CUSTOMER[11]</i>	0x0AC	Reserved for customer
<i>CUSTOMER[12]</i>	0x0B0	Reserved for customer
<i>CUSTOMER[13]</i>	0x0B4	Reserved for customer
<i>CUSTOMER[14]</i>	0x0B8	Reserved for customer
<i>CUSTOMER[15]</i>	0x0BC	Reserved for customer
<i>CUSTOMER[16]</i>	0x0C0	Reserved for customer
<i>CUSTOMER[17]</i>	0x0C4	Reserved for customer
<i>CUSTOMER[18]</i>	0x0C8	Reserved for customer
<i>CUSTOMER[19]</i>	0x0CC	Reserved for customer
<i>CUSTOMER[20]</i>	0x0D0	Reserved for customer
<i>CUSTOMER[21]</i>	0x0D4	Reserved for customer
<i>CUSTOMER[22]</i>	0x0D8	Reserved for customer
<i>CUSTOMER[23]</i>	0x0DC	Reserved for customer
<i>CUSTOMER[24]</i>	0x0E0	Reserved for customer
<i>CUSTOMER[25]</i>	0x0E4	Reserved for customer
<i>CUSTOMER[26]</i>	0x0E8	Reserved for customer
<i>CUSTOMER[27]</i>	0x0EC	Reserved for customer
<i>CUSTOMER[28]</i>	0x0F0	Reserved for customer
<i>CUSTOMER[29]</i>	0x0F4	Reserved for customer
<i>CUSTOMER[30]</i>	0x0F8	Reserved for customer
<i>CUSTOMER[31]</i>	0x0FC	Reserved for customer
<i>PSELRESET[0]</i>	0x200	Mapping of the nRESET function (see POWER chapter for details)
<i>PSELRESET[1]</i>	0x204	Mapping of the nRESET function (see POWER chapter for details)
<i>APPROTECT</i>	0x208	Access Port protection
<i>NFCPINS</i>	0x20C	Setting of pins dedicated to NFC functionality: NFC antenna or GPIO

### 14.1.1 NRFFW[0]

Address offset: 0x014

Reserved for Nordic firmware design

## 14.1.2 NRFFW[1]

Address offset: 0x018

Reserved for Nordic firmware design

### 14.1.3 NRFFW[2]

Address offset: 0x01C

Reserved for Nordic firmware design

#### 14.1.4 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
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	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																											

### 14.1.5 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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																											

## 14.1.6 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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																										

### 14.1.7 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.8 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.9 NRFFW[8]

Address offset: 0x034

Reserved for Nordic firmware design

### 14.1.10 NRFFW[9]

Address offset: 0x038

Reserved for Nordic firmware design

### 14.1.11 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.12 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.13 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.14 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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																											

### 14.1.15 NRFFW[14]

Address offset: 0x04C

Reserved for Nordic firmware design

## 14.1.16 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	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	1
Id	RW	Field	Value Id																														
A	RW	NRFHW	Value																											Description			
A	RW	NRFHW	Reserved for Nordic hardware design																														

## 14.1.17 NRFHW[1]

Address offset: 0x054

Reserved for Nordic hardware design

## 14.1.18 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
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	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																											

### 14.1.19 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## 14.1.20 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		
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	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																										

## 14.1.21 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		
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	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																											

## 14.1.22 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## 14.1.23 NRFHW[7]

Address offset: 0x06C

Reserved for Nordic hardware design

## 14.1.24 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## 14.1.25 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### 14.1.26 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## 14.1.27 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## **14.1.28 CUSTOMER[0]**

Address offset: 0x080

Reserved for customer

## 14.1.29 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### **14.1.30 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### **14.1.31 CUSTOMER[3]**

Address offset: 0x08C

Reserved for customer

## **14.1.32 CUSTOMER[4]**

Address offset: 0x090

Reserved for customer

### **14.1.33 CUSTOMER[5]**

Address offset: 0x094

Reserved for customer

### **14.1.34 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### **14.1.35 CUSTOMER[7]**

Address offset: 0x09C

Reserved for customer

## **14.1.36 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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																											

### **14.1.37 CUSTOMER[9]**

Address offset: 0x0A4

Reserved for customer

### **14.1.38 CUSTOMER[10]**

Address offset: 0x0A8

Reserved for customer

## **14.1.39 CUSTOMER[11]**

Address offset: 0x0AC

Reserved for customer

## **14.1.40 CUSTOMER[12]**

Address offset: 0x0B0

Reserved for customer

## **14.1.41 CUSTOMER[13]**

Address offset: 0x0B4

Reserved for customer

#### **14.1.42 CUSTOMER[14]**

Address offset: 0x0B8

Reserved for customer

#### **14.1.43 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

#### **14.1.44 CUSTOMER[16]**

Address offset: 0x0C0

Reserved for customer

## **14.1.45 CUSTOMER[17]**

Address offset: 0x0C4

Reserved for customer

## **14.1.46 CUSTOMER[18]**

Address offset: 0x0C8

Reserved for customer

#### **14.1.47 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	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																											

## **14.1.48 CUSTOMER[20]**

Address offset: 0x0D0

Reserved for customer

#### **14.1.49 CUSTOMER[21]**

Address offset: 0x0D4

Reserved for customer

## **14.1.50 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### **14.1.51 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## **14.1.52 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		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																												

### **14.1.53 CUSTOMER[25]**

Address offset: 0x0E4

Reserved for customer

## **14.1.54 CUSTOMER[26]**

Address offset: 0x0E8

Reserved for customer

## 14.1.55 CUSTOMER[27]

Address offset: 0x0EC

Reserved for customer

## **14.1.56 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	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## 14.1.57 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

### **14.1.58 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	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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																											

## **14.1.59 CUSTOMER[31]**

Address offset: 0x0FC

Reserved for customer

## 14.1.60 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 they don't, there will be no nRESET function exposed on a GPIO, and the device will always start independently of the levels present on any of the GPIOs.

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

### **14.1.61 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 they don't, there will be no nRESET function exposed on a GPIO, and the device will always start independently of the levels present on any of the GPIOs.

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

## 14.1.62 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				
<b>Reset 0xFFFFFFFF</b>	<b>1 1</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	PALL			Enable or disable Access Port protection. Any other value than 0xFF being written to this field will enable protection.  See <a href="#">Debug and trace</a> on page 72 for more information.
		Disabled		0xFF	Disable
		Enabled		0x00	Enable

### 14.1.63 NFCPINS

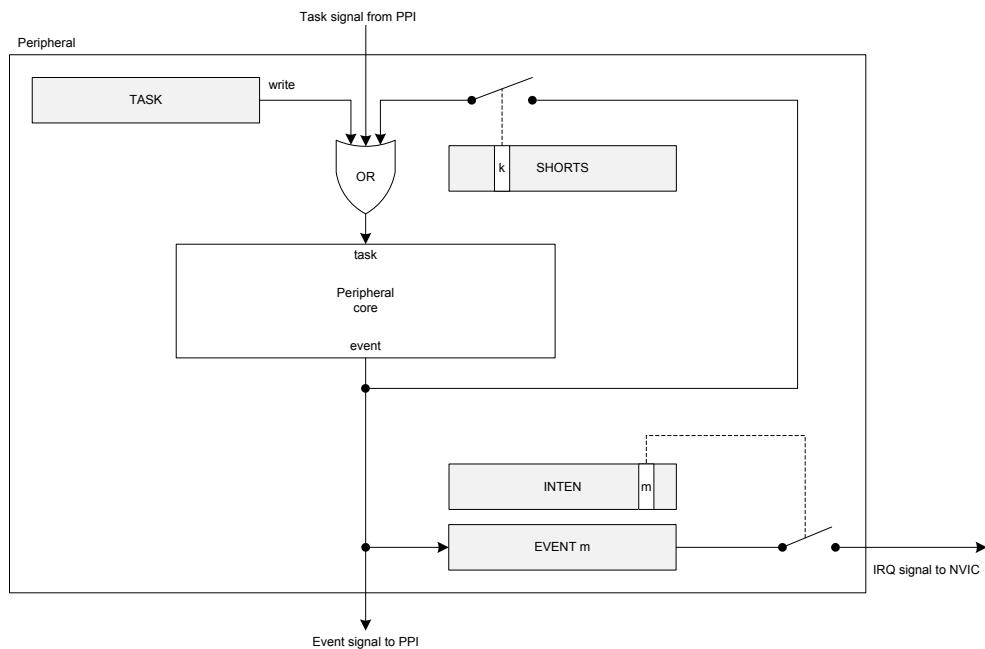
Address offset: 0x20C

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

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

# 15 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 10: Tasks, events, shortcuts, and interrupts**

## 15.1 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 24 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).

## 15.2 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.

For each of the rows in the following table, the instance ID listed is shared by the peripherals in the same row.

**Table 20: Peripherals sharing an ID**

Instance						
ID 2 (0x40002000)	UARTE	UART				
-						
ID 3 (0x40003000)	SPIM	SPIS	SPI	TWIM	TWIS	TWI
ID 4 (0x40004000)	SPIM	SPIS	SPI	TWIM	TWIS	TWI
ID 35 (0x40023000)	SPIM	SPIS	SPI			
-						
ID 15 (0x4000F000)	AAR	CCM				
-						
ID 19 (0x40013000)	COMP	LPCOMP				
-						
ID 20 (0x40014000)	SWI	EGU				
ID 21 (0x40015000)	SWI	EGU				
ID 22 (0x40016000)	SWI	EGU				
ID 23 (0x40017000)	SWI	EGU				
ID 24 (0x40018000)	SWI	EGU				
ID 25 (0x40019000)	SWI	EGU				

## 15.3 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.

## 15.4 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.

## 15.5 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 [Figure 10: Tasks, events, shortcuts, and interrupts](#) on page 68.

## 15.6 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 [Figure 10: Tasks, events, shortcuts, and interrupts](#) on page 68. 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'.

## 15.7 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.

## 15.8 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 [Figure 10: Tasks, events, shortcuts, and interrupts](#) on page 68.

### 15.8.1 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 immediately 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.

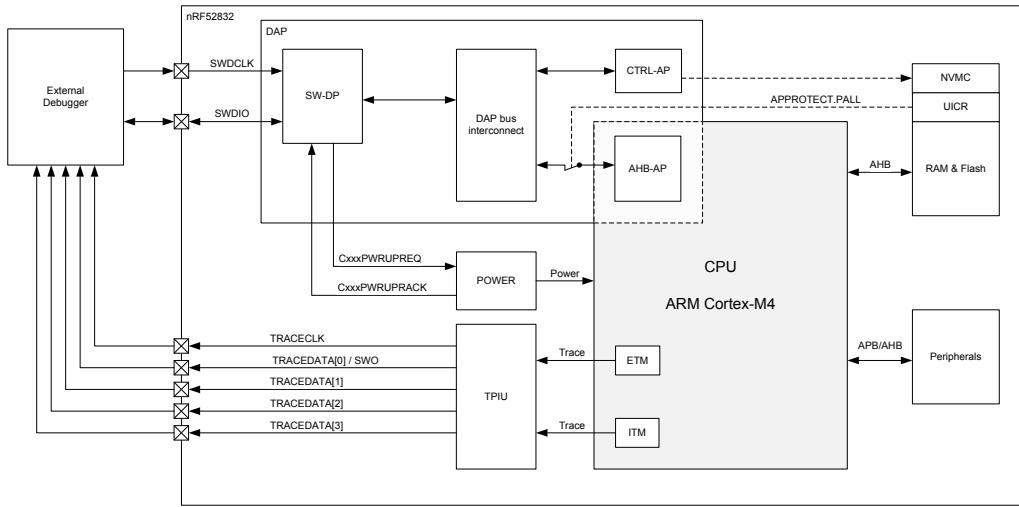
**Important:** To avoid an interrupt reoccurring before a new event has come, the program should perform a read from one of the peripheral registers, for example, the event register that has been cleared, or the INTENCLR register that has been used to disable the interrupt.

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.



# 16 Debug and trace

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



**Figure 11: Debug and trace overview**

The main features of the debug and trace system are:

- Two-pin Serial Wire Debug (SWD) interface
- Flash Patch and Breakpoint Unit (FPB) supports:
  - Two literal comparators
  - Six instruction comparators
- Data Watchpoint and Trace Unit (DWT)
  - Four comparators
- Instrumentation Trace Macrocell (ITM)
- Embedded Trace Macrocell (ETM)
- Trace Port Interface Unit (TPIU)
  - 4-bit parallel trace of ITM and ETM trace data
  - Serial Wire Output (SWO) trace of ITM data

## 16.1 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 [Figure 11: Debug and trace overview](#) on page 72.

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 73.

**Important:**

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

## 16.2 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 [APPROMT](#) on page 66 for more information about enabling access port protection.

This access port enables the following features:

- Soft reset, see [Reset](#) on page 82 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.

### 16.2.1 Registers

**Table 21: Register Overview**

Register	Offset	Description
<a href="#">RESET</a>	0x000	Soft reset triggered through CTRL-AP
<a href="#">ERASEALL</a>	0x004	Erase all
<a href="#">ERASEALLSTATUS</a>	0x008	Status register for the ERASEALL operation
<a href="#">APPROMTSTATUS</a>	0x00C	Status register for access port protection
<a href="#">IDR</a>	0x0FC	CTRL-AP Identification Register, IDR

#### 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																												

#### 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																											

#### ERASEALLSTATUS

Address offset: 0x008

Status register for the ERASEALL operation

# 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																																
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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																													

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	C	C	C	C	C	C	C	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x02880000</b>	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	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
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																											

## 16.3 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 85 will be set.

## 16.4 Real-time debug

The nRF52832 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.

## 16.5 Trace

The device supports ETM and ITM trace.

Trace data from the ETM and the ITM is sent to an external debugger via a 4-bit wide parallel trace port (TPIU), see TRACEDATA[0] through TRACEDATA[3] and TRACECLK in [Figure 11: Debug and trace overview](#) on page 72.

In addition to parallel trace, the TPIU supports serial trace via the Serial Wire Output (SWO) trace protocol.

Parallel and serial trace cannot be used at the same time.

ETM trace is only supported in parallel trace mode while ITM trace is supported in both parallel and serial trace modes.

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

TPIU's trace pins are multiplexed with GPIOs, and SWO and TRACEDATA[0] use the same GPIO, see [Pin assignments](#) on page 13 for more information.

Trace speed is configured in the [TRACECONFIG](#) on page 108 register.

The speed of the trace pins depends on the DRIVE setting of the GPIOs that the trace pins are multiplexed with, see [PIN\\_CNF\[14\]](#) on page 142, [PIN\\_CNF\[15\]](#) on page 143, [PIN\\_CNF\[16\]](#) on page 144, [PIN\\_CNF\[18\]](#) on page 145 and [PIN\\_CNF\[20\]](#) on page 146. Only S0S1 and H0H1 drives are suitable for debugging. S0S1 is the default DRIVE at reset. If parallel or serial trace port signals are not fast enough in the debugging conditions, all GPIOs in use for tracing should be set to high drive (H0H1). The user shall make sure that these GPIOs' DRIVE is not overwritten by software during the debugging session.

### 16.5.1 Electrical specification

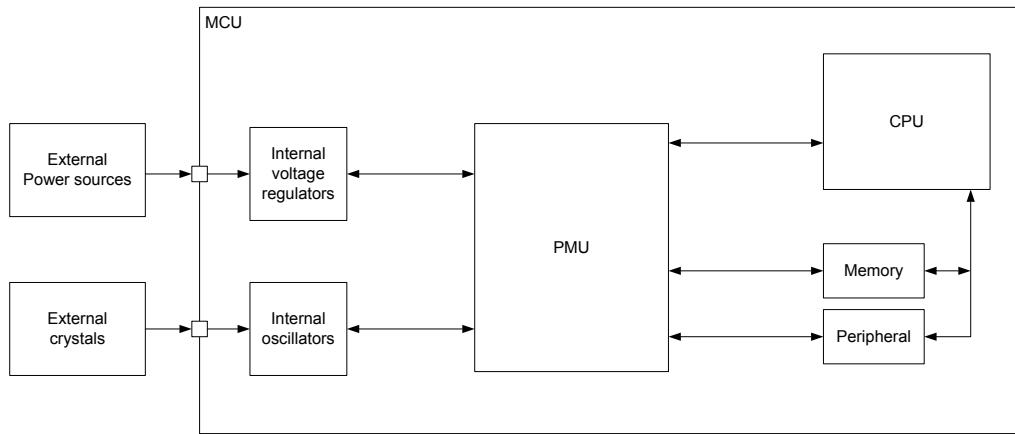
#### Trace port

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

# 17 Power and clock management

Power and clock management in nRF52832 is optimized for ultra-low power applications.

The core of the power and clock management system is the Power Management Unit (PMU) illustrated in [Figure 12: Power Management Unit](#) on page 76.



**Figure 12: Power Management Unit**

The user application is not required to actively control power and clock, since the PMU is able to automatically detect which resources are required by the different components in the system at any given time. The PMU will continuously optimize the system based on this information to achieve the lowest power consumption possible without user interaction.

## 17.1 Current consumption scenarios

As the system is being constantly tuned by the PMU, estimating the energy consumption of an application can be challenging if the designer is not able to do measurements on the hardware directly. See [Electrical specification](#) on page 76 for application scenarios showing average current drawn from the VDD supply.

Each scenario specifies a set of active operations and conditions applying to the given scenario. [Table 22: Current consumption scenarios, common conditions](#) on page 76 shows the conditions used for a scenario unless otherwise is stated in the scenario description.

**Table 22: Current consumption scenarios, common conditions**

Condition	Value
VDD	3 V
Temperature	25°C
CPU	WFI/WFE sleep
Peripherals	All idle
Clock	Not running
Regulator	DCDC

### 17.1.1 Electrical specification

#### Current consumption: Radio

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>RADIO_TX0</sub>	0 dBm TX @ 1 Mb/s Bluetooth Low Energy mode, Clock = HFXO		7.1		mA
I <sub>RADIO_TX1</sub>	-40 dBm TX @ 1 Mb/s Bluetooth Low Energy mode, Clock = HFXO		4.1		mA
I <sub>RADIO_RX0</sub>	Radio RX @ 1 Mb/s Bluetooth Low Energy mode, Clock = HFXO		6.5		mA

### Current consumption: Radio protocol configurations

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>S0</sub>	CPU running CoreMark from Flash, Radio 0 dBm TX @ 1 Mb/s Bluetooth Low Energy mode, Clock = HFXO, Cache enabled		9.6		mA
I <sub>S1</sub>	CPU running CoreMark from Flash, Radio RX @ 1 Mb/s Bluetooth Low Energy mode, Clock = HFXO, Cache enabled		9.0		mA

### Current consumption: Ultra-low power

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>ON_RAMOFF_EVENT</sub>	System ON, No RAM retention, Wake on any event		1.2		µA
I <sub>ON_RAMON_EVENT</sub>	System ON, Full RAM retention, Wake on any event		1.5		µA
I <sub>ON_RAMOFF_RTC</sub>	System ON, No RAM retention, Wake on RTC		1.9		µA
I <sub>OFF_RAMOFF_RESET</sub>	System OFF, No RAM retention, Wake on reset		0.3		µA
I <sub>OFF_RAMOFF_GPIO</sub>	System OFF, No RAM retention, Wake on GPIO		1.2		µA
I <sub>OFF_RAMOFF_LPCOMP</sub>	System OFF, No RAM retention, Wake on LPCOMP		1.9		µA
I <sub>OFF_RAMOFF_NFC</sub>	System OFF, No RAM retention, Wake on NFC field		0.7		µA
I <sub>OFF_RAMON_RESET</sub>	System OFF, Full 64 kB RAM retention, Wake on reset		0.7		µA

# 18 POWER — Power supply

This device has the following power supply features:

- On-chip LDO and DC/DC regulators
- Global System ON/OFF modes
- Individual RAM section power control for all system modes
- Analog or digital pin wakeup from System OFF
- Supervisor 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.

## 18.1 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 88 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 [Figure 14: DC/DC regulator setup](#) on page 79.

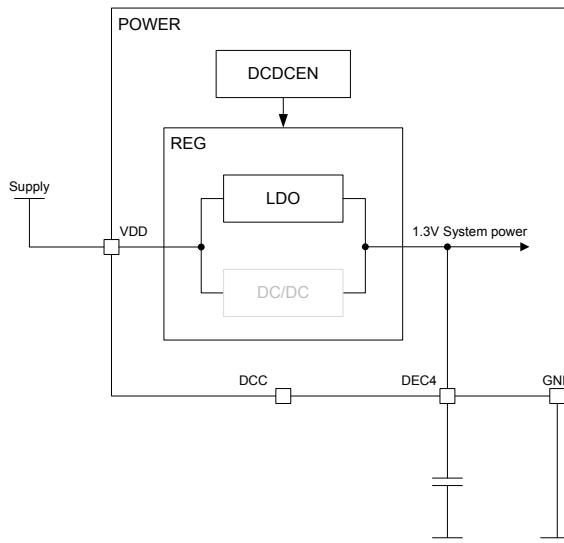
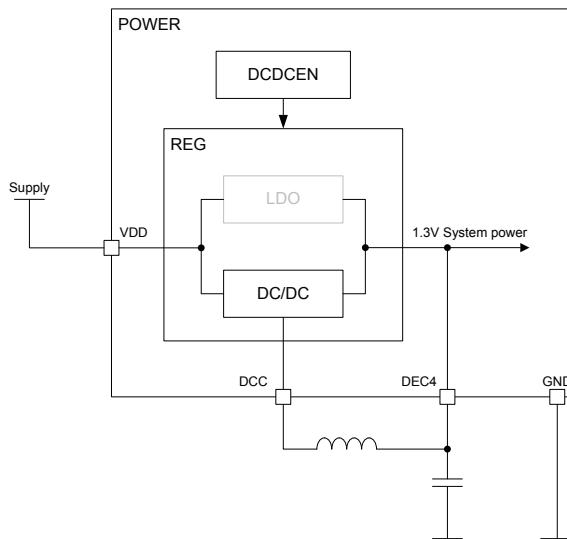


Figure 13: LDO regulator setup



**Figure 14: DC/DC regulator setup**

## 18.2 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 signals:

1. The DETECT signal, optionally generated by the GPIO peripheral
2. The ANADETECT signal, optionally generated by the LPCOMP module
3. The SENSE signal, optionally generated by the NFC module to “wake-on-field”
4. A reset

When the system wakes up from System OFF mode, it gets reset. For more details, see [Reset behavior](#) on page 83.

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.

### 18.2.1 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 and trace](#) on page 72 for more information. Required resources needed for debugging include the following key components: [Debug and trace](#) on page 72, [CLOCK — Clock control](#) on page 101, [POWER — Power supply](#) on page 78, [NVMC — Non-volatile memory controller](#) on page 29, 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.

## 18.3 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 85 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.

### 18.3.1 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.

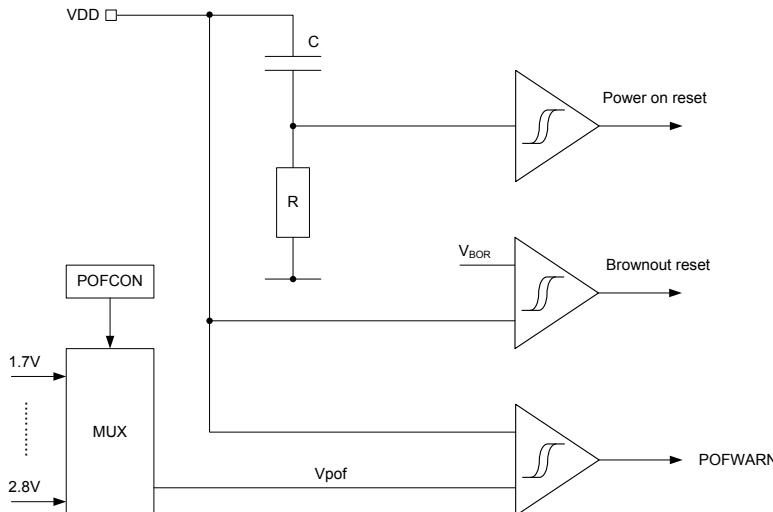
In low power mode the automatic power management system, described in [System ON mode](#) on page 80, ensures the most efficient supply option is chosen to save the most power. The advantage of having the lowest power possible will be at the cost of having varying CPU wakeup latency and PPI task response. 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.

## 18.4 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 [Figure 15: Power supply supervisor](#) on page 81.



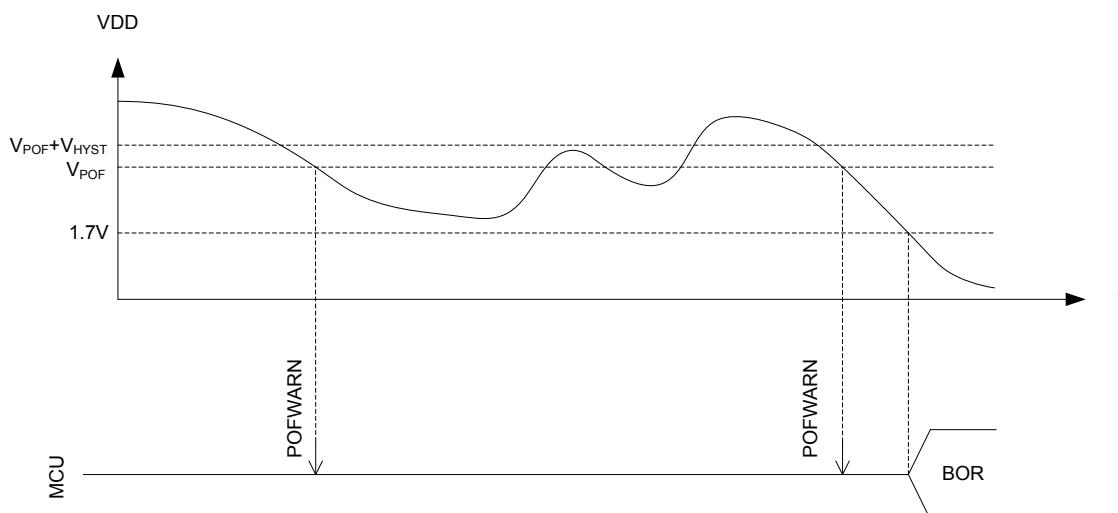
**Figure 15: Power supply supervisor**

#### 18.4.1 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 [Figure 16: Power-fail comparator \(BOR = Brownout reset\)](#) on page 81. The threshold  $V_{POF}$  is set in register [POFCON](#) on page 86. 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 29 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.

## 18.5 RAM sections

RAM section power control is used for retention in System OFF mode and for powering down unused sections in System ON mode.

Each RAM section can power up and down independently in both System ON and System OFF mode. See chapter [Memory](#) on page 23 for more information on RAM sections.

## 18.6 Reset

There are multiple sources that may trigger a reset.

After a reset has occurred, register [RESETREAS](#) can be read to determine which source generated the reset.

### 18.6.1 Power-on reset

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

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

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.

### 18.6.2 Pin reset

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

Pin reset is configured via the [PSELRESET\[0\]](#) and [PSELRESET\[1\]](#) registers.

**Note:** Pin reset is not available on all pins.

### 18.6.3 Wakeup from System OFF mode reset

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

The DAP is not reset following a wake up from System OFF mode if the device is in debug interface mode. Refer to chapter [Debug and trace](#) on page 72 for more information.

### 18.6.4 Soft reset

A soft reset is generated when the SYSRESETREQ bit of the Application Interrupt and Reset Control Register (AIRCR register) in the ARM® core is set.

Refer to [ARM documentation](#) for more details.

A soft reset can also be generated via the [RESET](#) on page 73 register in the CTRL-AP.

### 18.6.5 Watchdog reset

A Watchdog reset is generated when the watchdog times out.

Refer to chapter [WDT — Watchdog timer](#) on page 409 for more information.

### 18.6.6 Brown-out reset

The brown-out reset generator puts the system in reset state if the supply voltage drops below the brownout reset (BOR) threshold.

Refer to section [Power fail comparator](#) on page 99 for more information.

## 18.7 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.

## 18.8 Reset behavior

Reset source	Reset target								
	CPU	Peripherals	GPIO	Debug <sup>a</sup>	SWJ-DP	RAM	WDT	Retained registers	RESETREAS
CPU lockup <sup>6</sup>	x	x	x						
Soft reset	x	x	x						
Wakeup from System OFF mode reset	x	x		x <sup>7</sup>		x <sup>8</sup>			
Watchdog reset <sup>9</sup>	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.

## 18.9 Registers

**Table 23: Instances**

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

**Table 24: Register Overview**

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_SLEEPEENTER	0x114	CPU entered WFI/WFE sleep	
EVENTS_SLEEPEXIT	0x118	CPU exited WFI/WFE sleep	
<i>INTENSET</i>	0x304	Enable interrupt	
<i>INTENCLR</i>	0x308	Disable interrupt	
<i>RESETREAS</i>	0x400	Reset reason	
<i>RAMSTATUS</i>	0x428	RAM status register	Deprecated
<i>SYSTEMOFF</i>	0x500	System OFF register	
<i>POFCON</i>	0x510	Power failure comparator configuration	
<i>GPREGRET</i>	0x51C	General purpose retention register	
<i>GPREGRET2</i>	0x520	General purpose retention register	
<i>RAMON</i>	0x524	RAM on/off register (this register is retained)	Deprecated
<i>RAMONB</i>	0x554	RAM on/off register (this register is retained)	Deprecated
<i>DCDCEN</i>	0x578	DC/DC enable register	
<i>RAM[0].POWER</i>	0x900	RAM0 power control register	

<sup>a</sup> All debug components excluding SWJ-DP. See [Debug and trace](#) on page 72 chapter for more information about the different debug components in the system.

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

<sup>7</sup> The Debug components will not be reset if the device is in debug interface mode.

<sup>8</sup> 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.

<sup>9</sup> Watchdog reset is not available in System OFF.

Register	Offset	Description
<i>RAM[0].POWERSET</i>	0x904	RAM0 power control set register
<i>RAM[0].POWERCLR</i>	0x908	RAM0 power control clear register
<i>RAM[1].POWER</i>	0x910	RAM1 power control register
<i>RAM[1].POWERSET</i>	0x914	RAM1 power control set register
<i>RAM[1].POWERCLR</i>	0x918	RAM1 power control clear register
<i>RAM[2].POWER</i>	0x920	RAM2 power control register
<i>RAM[2].POWERSET</i>	0x924	RAM2 power control set register
<i>RAM[2].POWERCLR</i>	0x928	RAM2 power control clear register
<i>RAM[3].POWER</i>	0x930	RAM3 power control register
<i>RAM[3].POWERSET</i>	0x934	RAM3 power control set register
<i>RAM[3].POWERCLR</i>	0x938	RAM3 power control clear register
<i>RAM[4].POWER</i>	0x940	RAM4 power control register
<i>RAM[4].POWERSET</i>	0x944	RAM4 power control set register
<i>RAM[4].POWERCLR</i>	0x948	RAM4 power control clear register
<i>RAM[5].POWER</i>	0x950	RAM5 power control register
<i>RAM[5].POWERSET</i>	0x954	RAM5 power control set register
<i>RAM[5].POWERCLR</i>	0x958	RAM5 power control clear register
<i>RAM[6].POWER</i>	0x960	RAM6 power control register
<i>RAM[6].POWERSET</i>	0x964	RAM6 power control set register
<i>RAM[6].POWERCLR</i>	0x968	RAM6 power control clear register
<i>RAM[7].POWER</i>	0x970	RAM7 power control register
<i>RAM[7].POWERSET</i>	0x974	RAM7 power control set register
<i>RAM[7].POWERCLR</i>	0x978	RAM7 power control clear register

## 18.9.1 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			
<b>A</b>	RW	POFWARN																																
<b>B</b>	RW	SLEEPENTER																																
<b>C</b>	RW	SLEEPEXIT																																

## 18.9.2 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			
<b>Reset 0x00000000</b>		<b>0 0</b>			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	POFWARN			Write '1' to Disable interrupt for POFWARN event
					See <a href="#">EVENTS_POFWARN</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
B	RW	SLEEPENTER			Write '1' to Disable interrupt for SLEEPENTER event
					See <a href="#">EVENTS_SLEEPENTER</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
C	RW	SLEEPEXIT			Write '1' to Disable interrupt for SLEEPEXIT event
					See <a href="#">EVENTS_SLEEPEXIT</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled

### 18.9.3 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		H G F E D C B A			
<b>Reset 0x00000000</b>		<b>0 0</b>			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	RESETPIN			Reset from pin-reset detected
		NotDetected	0		Not detected
		Detected	1		Detected
B	RW	DOG			Reset from watchdog detected
		NotDetected	0		Not detected
		Detected	1		Detected
C	RW	SREQ			Reset from soft reset detected
		NotDetected	0		Not detected
		Detected	1		Detected
D	RW	LOCKUP			Reset from CPU lock-up detected
		NotDetected	0		Not detected
		Detected	1		Detected
E	RW	OFF			Reset due to wake up from System OFF mode when wakeup is triggered from DETECT signal from GPIO
		NotDetected	0		Not detected
		Detected	1		Detected
F	RW	LPCOMP			Reset due to wake up from System OFF mode when wakeup is triggered from ANADETECT signal from LPCOMP
		NotDetected	0		Not detected
		Detected	1		Detected
G	RW	DIF			Reset due to wake up from System OFF mode when wakeup is triggered from entering into debug interface mode
		NotDetected	0		Not detected
		Detected	1		Detected

#### **18.9.4 RAMSTATUS ( Deprecated )**

Address offset: 0x428

## RAM status register

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

## **18.9.5 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																															
<b>Reset 0x00000000</b>																																
<b>Id RW Field Value Id Value Description</b>																																
A	W	SYSTEMOFF																										Enable System OFF mode				
		Enter																										Enable System OFF mode				

## 18.9.6 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	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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																												

Bit number	31 30 29 28 27 26 25 24 23 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
<b>Id RW Field Value Id Value Description</b>	
	Enabled 1 Enable
B RW THRESHOLD	
V17	4 Power failure comparator threshold setting
V18	5 Set threshold to 1.7 V
V19	6 Set threshold to 1.8 V
V20	7 Set threshold to 1.9 V
V21	8 Set threshold to 2.0 V
V22	9 Set threshold to 2.1 V
V23	10 Set threshold to 2.2 V
V24	11 Set threshold to 2.3 V
V25	12 Set threshold to 2.4 V
V26	13 Set threshold to 2.5 V
V27	14 Set threshold to 2.6 V
V28	15 Set threshold to 2.7 V
	Set threshold to 2.8 V

## 18.9.7 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
<b>Id RW Field Value Id Value Description</b>	
A RW GPREGRET	General purpose retention register
	This register is a retained register

## 18.9.8 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
<b>Id RW Field Value Id Value Description</b>	
A RW GPREGRET	General purpose retention register
	This register is a retained register

## 18.9.9 RAMON ( Deprecated )

Address offset: 0x524

RAM on/off register (this register is retained)

Since this register is deprecated the following substitutions have been made: RAM block 0 is equivalent to a block comprising RAM0.S0 and RAM0.S1 and RAM block 1 is equivalent to a block comprising RAM1.S0 and RAM1.S1. For new designs it is recommended to use the POWER.RAM-0.POWER and its sibling registers instead.

Bit number	31 30 29 28 27 26 25 24 23 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 0x00000003	0 1 1
<b>Id RW Field Value Id Value Description</b>	
A RW ONRAMO	Keep RAM block 0 on or off in system ON Mode
RAM0Off	0 Off

Bit number		31 30 29 28 27 26 25 24 23 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			
<b>Reset 0x00000003</b>		0 1 1			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
B RW ONRAM1		RAM0On	1	On	Keep RAM block 1 on or off in system ON Mode
		RAM1Off	0	Off	
		RAM1On	1	On	
C RW OFFRAM0		RAM0Off	0	Off	Keep retention on RAM block 0 when RAM block is switched off
		RAM0On	1	On	
		RAM1Off	0	Off	Keep retention on RAM block 1 when RAM block is switched off
D RW OFFRAM1		RAM1On	1	On	

### 18.9.10 RAMONB ( Deprecated )

Address offset: 0x554

RAM on/off register (this register is retained)

Since this register is deprecated the following substitutions have been made: RAM block 2 is equivalent to a block comprising RAM2.S0 and RAM2.S1 and RAM block 3 is equivalent to a block comprising RAM3.S0 and RAM3.S1. For new designs it is recommended to use the POWER.RAM-0.POWER and its sibling registers instead.

Bit number		31 30 29 28 27 26 25 24 23 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			
<b>Reset 0x00000003</b>		0 1 1			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A RW ONRAM2		RAM2On	1	On	Keep RAM block 2 on or off in system ON Mode
		RAM2Off	0	Off	
		RAM3On	1	On	
B RW ONRAM3		RAM3Off	0	Off	Keep RAM block 3 on or off in system ON Mode
		RAM3On	1	On	
		RAM2Off	0	Off	Keep retention on RAM block 2 when RAM block is switched off
C RW OFFRAM2		RAM2On	1	On	
		RAM3Off	0	Off	Keep retention on RAM block 3 when RAM block is switched off
		RAM3On	1	On	
D RW OFFRAM3		RAM3Off	0	Off	
		RAM2On	1	On	

### 18.9.11 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			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A RW DCDCEN		Disabled	0	Disable	Enable or disable DC/DC converter
		Enabled	1	Enable	

### 18.9.12 RAM[0].POWER

Address offset: 0x900

RAM0 power control register

### **18.9.13 RAM[0].POWERSET**

Address offset: 0x904

## RAM0 power control set register

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

## 18.9.14 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	
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	W	SOPOWER			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																											

Bit number	31	30	29	28	27	26	25	24	23	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			
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
		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																											

## 18.9.15 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	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id																										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						
																												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						
																												On						
C RW SORETENTION																												Keep retention on RAM section S0 when RAM section is in OFF						
																												Off						
																												On						
D RW S1RETENTION																												Keep retention on RAM section S1 when RAM section is in OFF						
																												Off						
																												On						

## 18.9.16 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					
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value													Description															
A	W	SOPOWER		On													Keep RAM section S0 of RAM1 on or off in System ON mode															
			On	1													On															
B	W	S1POWER		On													Keep RAM section S1 of RAM1 on or off in System ON mode															
			On	1													On															
C	W	SORETENTION															Keep retention on RAM section S0 when RAM section is switched off															
			On	1													On															

## 18.9.17 RAM[1].POWERCLR

Address offset: 0x918

## RAM1 power control clear register

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

## 18.9.18 RAM[2].POWER

Address offset: 0x920

## RAM2 power control register

## **18.9.19 RAM[2].POWERSET**

Address offset: 0x924

## RAM2 power control set register

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

## **18.9.20 RAM[2].POWERCLR**

Address offset: 0x928

## RAM2 power control clear register

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

## 18.9.21 RAM[3].POWER

Address offset: 0x930

## RAM3 power control register

## 18.9.22 RAM[3].POWERSET

Address offset: 0x934

## RAM3 power control set register

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

## 18.9.23 RAM[3].POWERCLR

Address offset: 0x938

## RAM3 power control clear register

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

## 18.9.24 RAM[4].POWER

Address offset: 0x940

## RAM4 power control register

## 18.9.25 RAM[4].POWERSET

Address offset: 0x944

## RAM4 power control set register

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

## 18.9.26 RAM[4].POWERCLR

Address offset: 0x948

## RAM4 power control clear register

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

## 18.9.27 RAM[5].POWER

Address offset: 0x950

## RAM5 power control register

## **18.9.28 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					
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	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																													

## 18.9.29 RAM[5].POWERCLR

Address offset: 0x958

## RAM5 power control clear register

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

## 18.9.30 RAM[6].POWER

Address offset: 0x960

## RAM6 power control register

### **18.9.31 RAM[6].POWERSET**

Address offset: 0x964

## RAM6 power control set register

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

### **18.9.32 RAM[6].POWERCLR**

Address offset: 0x968

## RAM6 power control clear register

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

## 18.9.33 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	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value																										Description					
																											RAM sections are always retained when ON, but can also be retained when OFF dependent on the settings in SORETENTION.							
																											All RAM sections will be OFF in System OFF mode.							
																											Off							
																											0							
																											Off							
																											On							
																											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							
																											On							
																											Keep retention on RAM section S0 when RAM section is in OFF							
																											Off							
																											0							
																											On							
																											Keep retention on RAM section S1 when RAM section is in OFF							
																											Off							
																											0							
																											On							

## **18.9.34 RAM[7].POWERSET**

Address offset: 0x974

## RAM7 power control set register

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

### **18.9.35 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				
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																												
A	W	SOPOWER			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																												

Bit number	31	30	29	28	27	26	25	24	23	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								
<b>Reset 0x0000FFFF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value	Id	Value	Description																										
C	W	SORETENTION				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																										

## **18.10 Electrical specification**

### 18.10.1 Current consumption, sleep

Symbol	Description	Min.	Typ.	Max.	Units
$I_{OFF}$	System OFF current, no RAM retention	0.3			$\mu A$
$I_{ON}$	System ON base current, no RAM retention		1.2		$\mu A$
$I_{RAM}$	Additional RAM retention current per 4 KB RAM section		20		nA

## 18.10.2 Device startup times

Symbol	Description	Min.	Typ.	Max.	Units
t <sub>POR</sub>	Time in Power on Reset after VDD reaches 1.7 V for all supply voltages and temperatures. Dependent on supply rise time. <sup>10</sup>				
t <sub>POR,10us</sub>	VDD rise time 10us		1		ms
t <sub>POR,10ms</sub>	VDD rise time 10ms		9		ms
t <sub>POR,60ms</sub>	VDD rise time 60ms		23		ms
t <sub>PINR</sub>	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) <sup>11</sup> : t=5RC, R = 13kOhm				
t <sub>PINR,500nF</sub>	C = 500nF		32.5		ms
t <sub>PINR,10uF</sub>	C = 10uF		650		ms
t <sub>R2ON</sub>	Time from reset to ON (CPU execute)				
t <sub>R2ON,NOTCONF</sub>	If reset pin not configured	tPOR			ms
t <sub>R2ON,CONF</sub>	If reset pin configured	tPOR + tPINR			ms
t <sub>OFF2ON</sub>	Time from OFF to CPU execute	16.5			μs
t <sub>IDLE2CPU</sub>	Time from IDLE to CPU execute	3.0			μs
t <sub>EVTSET,CL1</sub>	Time from HW event to PPI event in Constant Latency System  ON mode	0.0625			μs
t <sub>EVTSET,CLO</sub>	Time from HW event to PPI event in Low Power System ON  mode	0.0625			μs

### **18.10.3 Power fail comparator**

Symbol	Description	Min.	Typ.	Max.	Units
$I_{POF}$	Current consumption when enabled <sup>12</sup>		<4		µA
$V_{POF}$	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	%

<sup>10</sup> 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.

<sup>11</sup> To decrease maximum time a device could hold in reset, a strong external pullup resistor can be used.

<sup>12</sup> To save power, POF will not operate nor consume in System OFF, or while HFCLK is not running, even if left enabled by software

Symbol	Description	Min.	Typ.	Max.	Units
$V_{POFHYST}$	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	1.5		1.7	V

# 19 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 +/-250 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

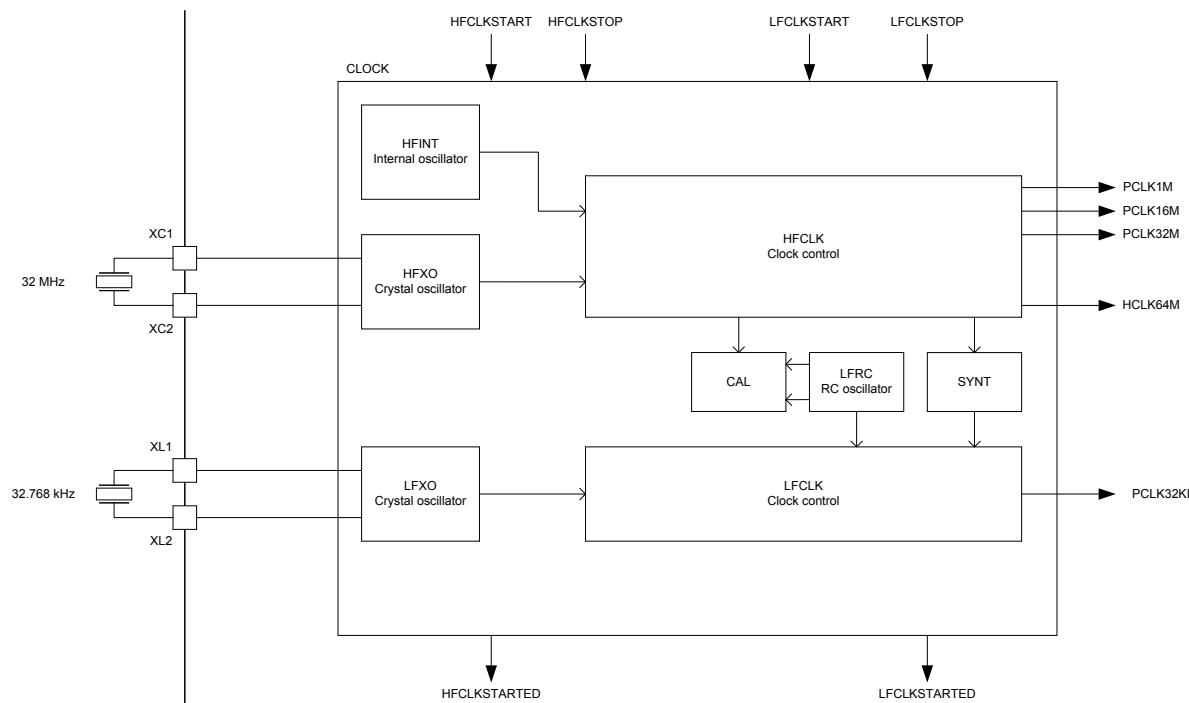


Figure 17: Clock control

## 19.1 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 [Figure 17: Clock control](#) on page 101.

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.

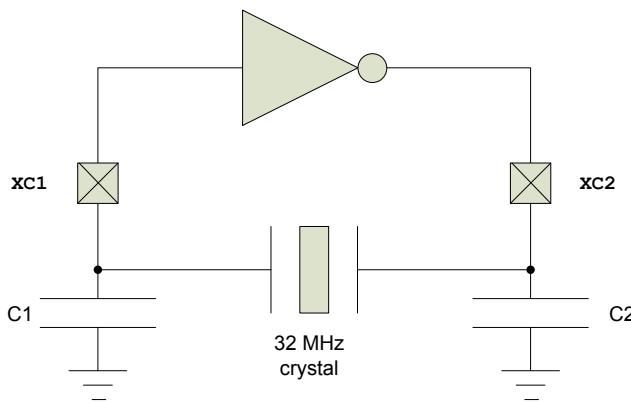
The HFXO must be running to use the RADIO, NFC module or the calibration mechanism associated with the 32.768 kHz RC oscillator.

### 19.1.1 64 MHz crystal oscillator (HFXO)

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

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

*Figure 18: Circuit diagram of the 64 MHz crystal oscillator* on page 102 shows how the 32 MHz crystal is connected to the 64 MHz crystal oscillator.



**Figure 18: Circuit diagram of the 64 MHz crystal oscillator**

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

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

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

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

$C1$  and  $C2$  are ceramic SMD capacitors connected between each crystal terminal and ground. For more information, see [Reference circuitry](#) on page 545.  $C_{pcb1}$  and  $C_{pcb2}$  are stray capacitances on the PCB.  $C_{pin}$  is the pin input capacitance on the  $XC1$  and  $XC2$  pins. See table [64 MHz crystal oscillator \(HFXO\)](#) on page 109. The load capacitors  $C1$  and  $C2$  should have the same value.

For reliable operation, the crystal load capacitance, shunt capacitance, equivalent series resistance, and drive level must comply with the specifications in table [64 MHz crystal oscillator \(HFXO\)](#) on page 109. It is recommended to use a crystal with lower than maximum load capacitance and/or shunt capacitance. A low load capacitance will reduce both start up time and current consumption.

## 19.2 LFCLK clock controller

The system supports several low frequency clock sources.

As illustrated in [Figure 17: Clock control](#) on page 101, the system supports the following low frequency clock sources:

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

The LFCLK clock is started by first selecting the preferred clock source in register [LFCLKSRC](#) on page 108 and then triggering the LFCLKSTART task. If the LFXO is selected as the clock source, the LFCLK will initially start running from the 32.768 kHz LFRC while the LFXO is starting up and automatically switch to using the LFXO once this oscillator is running. The LFCLKSTARTED event will be generated when the LFXO has been started.

The LFCLK clock is stopped by triggering the LFCLKSTOP task.

It is not allowed to write to register [LFCLKSRC](#) on page 108 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 107 indicates a 'LFCLK running' state.

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

### 19.2.1 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 109 for details on the default and calibrated accuracy of the LFRC oscillator.

The LFRC oscillator does not require additional external components.

### 19.2.2 Calibrating the 32.768 kHz RC oscillator

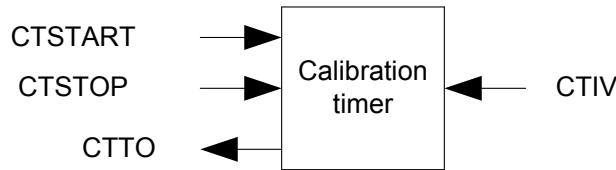
After the 32.768 kHz RC oscillator is started and running, it can be calibrated by triggering the CAL task. In this case, the HFCLK will be temporarily switched on and used as a reference.

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.

### 19.2.3 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.

### 19.2.4 32.768 kHz crystal oscillator (LFXO)

For higher LFCLK accuracy (when better than +/- 250 ppm accuracy is required), the low frequency crystal oscillator (LFXO) must be used.

The following external clock sources are supported:

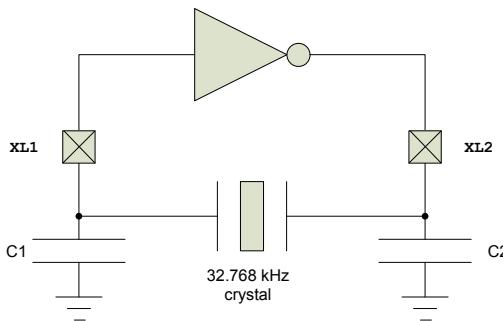
- Low swing clock signal applied to the *XL1* pin. The *XL2* pin shall then be grounded.
- Rail-to-rail clock signal applied to the *XL1* pin. The *XL2* pin shall then be grounded or left unconnected.

The [LFCLKSRC](#) on page 108 register controls the clock source, and its allowed swing. The truth table for various situations is as follows:

**Table 25: LFCLKSRC configuration depending on clock source**

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

To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet. [Figure 20: Circuit diagram of the 32.768 kHz crystal oscillator](#) on page 104 shows the LFXO circuitry.



**Figure 20: Circuit diagram of the 32.768 kHz crystal oscillator**

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

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

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

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

$C1$  and  $C2$  are ceramic SMD capacitors connected between each crystal terminal and ground.  $C_{pcb1}$  and  $C_{pcb2}$  are stray capacitances on the PCB.  $C_{pin}$  is the pin input capacitance on the XC1 and XC2 pins (see [32.768 kHz crystal oscillator \(LFXO\)](#) on page 109). The load capacitors  $C1$  and  $C2$  should have the same value.

For more information, see [Reference circuitry](#) on page 545.

### 19.2.5 32.768 kHz synthesized from HFCLK (LFSYNT)

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

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

## 19.3 Registers

**Table 26: Instances**

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

**Table 27: Register Overview**

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_CSTOP	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_CTOO	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	
LFCLKSRCOPY	0x41C	Copy of LFCLKSRC register, set when LFCLKSTART task was triggered	
LFCLKSRC	0x518	Clock source for the LFCLK	
CTIV	0x538	Calibration timer interval	
TRACECONFIG	0x55C	Clocking options for the Trace Port debug interface	Retained

### 19.3.1 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		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	HFCLKSTARTED				Write '1' to Enable interrupt for HFCLKSTARTED event																										
						See <a href="#">EVENTS_HFCLKSTARTED</a>																										
		Set	1			Enable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
B	RW	LFCLKSTARTED				Write '1' to Enable interrupt for LFCLKSTARTED event																										
						See <a href="#">EVENTS_LFCLKSTARTED</a>																										
		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			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
C	RW	DONE			Write '1' to Enable interrupt for DONE event  See <a href="#">EVENTS_DONE</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CTTO			Write '1' to Enable interrupt for CTTO event  See <a href="#">EVENTS_CTTO</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

### 19.3.2 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			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	HFCLKSTARTED			Write '1' to Disable interrupt for HFCLKSTARTED event  See <a href="#">EVENTS_HFCLKSTARTED</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	LFCLKSTARTED			Write '1' to Disable interrupt for LFCLKSTARTED event  See <a href="#">EVENTS_LFCLKSTARTED</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to Disable interrupt for DONE event  See <a href="#">EVENTS_DONE</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CTTO			Write '1' to Disable interrupt for CTTO event  See <a href="#">EVENTS_CTTO</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

### 19.3.3 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			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>R</b>	<b>STATUS</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	R	STATUS			HFCLKSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

### 19.3.4 HFCLKSTAT

Address offset: 0x40C

## HFCLK status

### **19.3.5 LFCLKRUN**

Address offset: 0x414

Status indicating that LFCLKSTART task has been triggered

### 19.3.6 LFCLKSTAT

Address offset: 0x418

## LFCLK status

### **19.3.7 LFCLKSRCCOPY**

Address offset: 0x41C

Copy of LFCLKSRC register, set when LFCLKSTART task was triggered

### 19.3.8 LFCLKSRC

Address offset: 0x518

## Clock source for the LFCLK

### **19.3.9 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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.																											

### 19.3.10 TRACECONFIG

Address offset: 0x55C

## Clocking options for the Trace Port debug interface

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

## 19.4 Electrical specification

### 19.4.1 64 MHz internal oscillator (HFINT)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM\_HFINT}}$	Nominal output frequency	64			MHz
$f_{\text{TOL\_HFINT}}$	Frequency tolerance		<±1.5	<±6	%
$I_{\text{HFINT}}$	Run current	60			µA
$I_{\text{START\_HFINT}}$	Average startup current		$I_{\text{HFINT}}$		µA
$t_{\text{START\_HFINT}}$	Startup time	3			us

### 19.4.2 64 MHz crystal oscillator (HFXO)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM\_HFXO}}$	Nominal output frequency	64			MHz
$f_{\text{XTAL\_HFXO}}$	External crystal frequency	32			MHz
$f_{\text{TOL\_HFXO}}$	Frequency tolerance requirement for 2.4 GHz proprietary radio applications		±60		ppm
$f_{\text{TOL\_HFXO\_BLE}}$	Frequency tolerance requirement, Bluetooth low energy applications		±40		ppm
$C_L_{\text{HFXO}}$	Load capacitance	12			pF
$C_0_{\text{HFXO}}$	Shunt capacitance	7			pF
$R_S_{\text{HFXO\_7PF}}$	Equivalent series resistance $C_0 = 7 \text{ pF}$	60			ohm
$R_S_{\text{HFXO\_5PF}}$	Equivalent series resistance $C_0 = 5 \text{ pF}$	80			ohm
$R_S_{\text{HFXO\_3PF}}$	Equivalent series resistance $C_0 = 3 \text{ pF}$	100			ohm
$P_D_{\text{HFXO}}$	Drive level	100			uW
$C_{\text{PIN\_HFXO}}$	Input capacitance XC1 and XC2	4			pF
$I_{\text{STBY\_X32M}}$	Core standby current <sup>13</sup>	50			µA
$I_{\text{HFXO}}$	Run current	250			µA
$I_{\text{START\_HFXO}}$	Average startup current, first 1 ms	0.4			mA
$t_{\text{START\_HFXO}}$	Startup time	0.36			ms

### 19.4.3 32.768 kHz RC oscillator (LFRC)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM\_LFRC}}$	Nominal frequency	32.768			kHz
$f_{\text{TOL\_LFRC}}$	Frequency tolerance		±2		%
$f_{\text{TOL\_CAL\_LFRC}}$	Frequency tolerance for LFRC after calibration <sup>14</sup>		±250		ppm
$I_{\text{LFRC}}$	Run current for 32.768 kHz RC oscillator	0.6	1		µA
$t_{\text{START\_LFRC}}$	Startup time for 32.768 kHz RC oscillator	600			us

### 19.4.4 32.768 kHz crystal oscillator (LFXO)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM\_LFXO}}$	Crystal frequency	32.768			kHz
$f_{\text{TOL\_LFXO\_BLE}}$	Frequency tolerance requirement for BLE stack		±250		ppm
$f_{\text{TOL\_LFXO\_ANT}}$	Frequency tolerance requirement for ANT stack		±50		ppm
$C_L_{\text{LFXO}}$	Load capacitance	12.5			pF
$C_0_{\text{LFXO}}$	Shunt capacitance	2			pF
$R_S_{\text{LFXO}}$	Equivalent series resistance	100			kohm
$P_D_{\text{LFXO}}$	Drive level	1			uW
$C_{\text{pin}}$	Input capacitance on XL1 and XL2 pads	4			pF
$I_{\text{LFXO}}$	Run current for 32.768 kHz crystal oscillator	0.25			µA

<sup>13</sup> Current drawn if HFXO is forced on through for instance using the low latency power mode.

<sup>14</sup> Constant temperature within ±0.5 °C and calibration performed at least every 8 seconds

Symbol	Description	Min.	Typ.	Max.	Units
$t_{START\_LFXO}$	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

### 19.4.5 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 HFCLK tolerance <sup>15</sup>		8		ppm
$I_{LFSYNT}$	Run current for synthesized 32.768 kHz	100			$\mu$ A
$t_{START\_LFSYNT}$	Startup time for synthesized 32.768 kHz		100		us

<sup>15</sup> Frequency tolerance will be derived from the HFCLK source clock plus the LFSYNT tolerance

## 20 GPIO — General purpose input/output

The general purpose input/output (GPIO) is organized as one port with up to 32 I/Os (dependent on package) enabling access and control of up to 32 pins through one port. Each GPIO can be accessed individually.

GPIO has the following user-configurable features:

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

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

The following parameters can be configured through these registers:

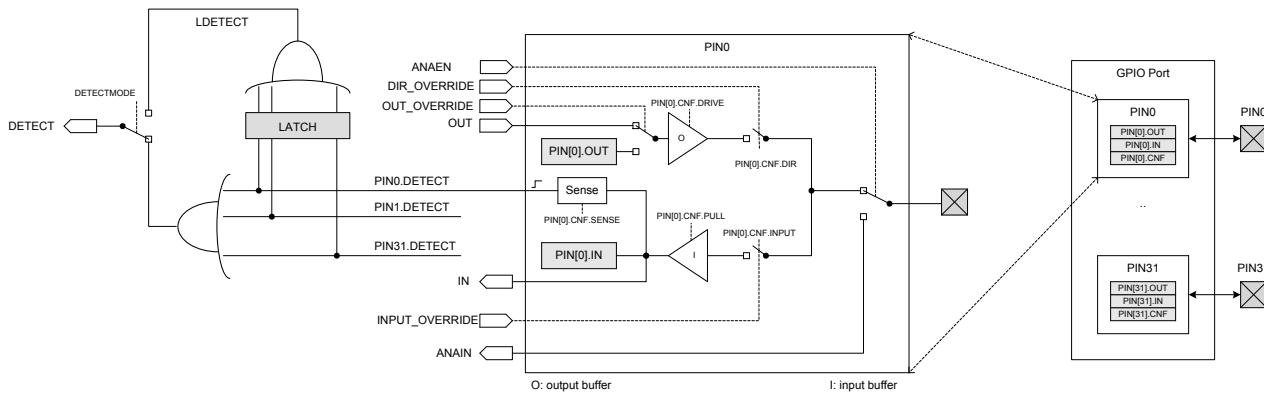
- Direction
- Drive strength
- Enabling of pull-up and pull-down resistors
- Pin sensing
- Input buffer disconnect
- Analog input (for selected pins)

The `PIN_CNF` registers are retained registers. See [POWER — Power supply](#) on page 78 chapter for more information about retained registers.

### 20.1 Pin configuration

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

When the correct level is detected on any such configured pin, the sense mechanism will set the DETECT signal high. Each pin has a separate DETECT signal, and the default behaviour, as defined by the DETECTMODE register, is that the DETECT signal from all pins in the GPIO Port are combined into a common DETECT signal that is routed throughout the system, which then can be utilized by other peripherals, see [Figure 21: GPIO Port and the GPIO pin details](#) on page 112. This mechanism is functional in both ON and OFF mode.



**Figure 21: GPIO Port and the GPIO pin details**

[Figure 21: GPIO Port and the GPIO pin details](#) on page 112 illustrates the GPIO port containing 32 individual pins, where **PIN0** is illustrated in more detail as a reference. All the signals on the left side of the illustration are used by other peripherals in the system, and therefore, are not directly available to the CPU.

Make sure that a pin is in a level that cannot trigger the sense mechanism before enabling it. Detect will go high immediately if the sense condition configured in the PIN\_CNF registers is met when the sense mechanism is enabled. This will trigger a PORT event if the DETECT signal was low before enabling the sense mechanism. See [GPIOE — GPIO tasks and events](#) on page 157.

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

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

When a pin's PINx.DETECT signal goes high, a flag will be set in the LATCH register, e.g. when the PIN0.DETECT signal goes high, bit 0 in the LATCH register will be set to '1'.

The LATCH register will only be cleared if the CPU explicitly clears it by writing a '1' to the bit that shall be cleared, i.e. the LATCH register will not be affected by a PINx.DETECT signal being set low.

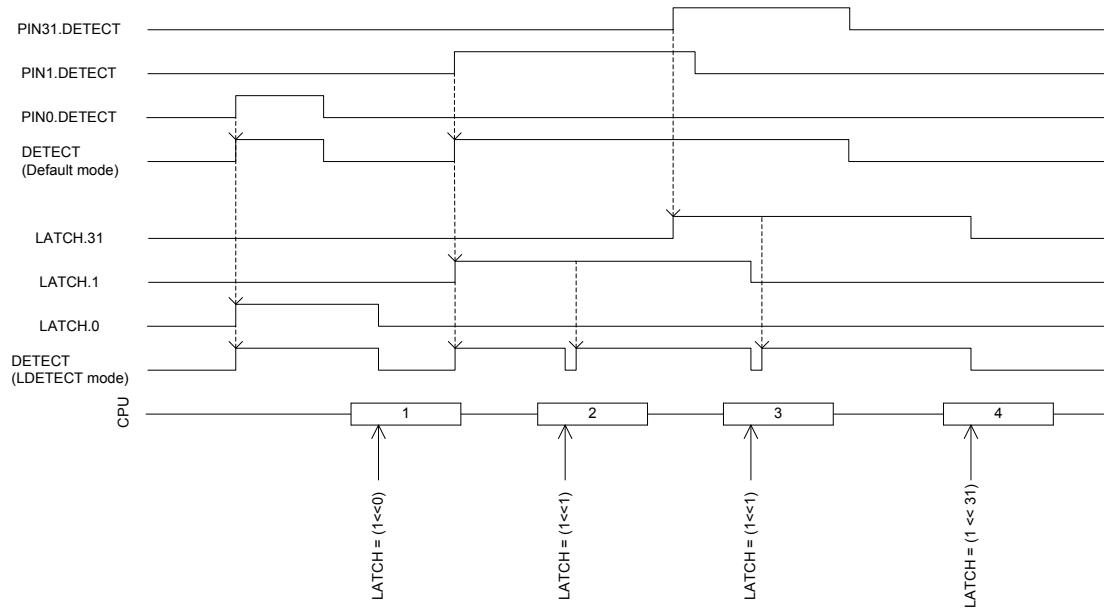
If the CPU performs a clear operation on a bit in the LATCH register when the associated PINx.DETECT signal is high, the bit in the LATCH register will not be cleared.

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

If one or more bits in the LATCH register are '1' after the CPU has performed a clear operation on the LATCH registers, a rising edge will be generated on the LDETECT signal, this is illustrated in [Figure 22: DETECT signal behavior](#) on page 113.

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

The LDETECT signal is by default not connected to the GPIO port's DETECT signal, but via the DETECTMODE register it is possible to change the behaviour of the GPIO port's DETECT signal from the default behaviour described above to instead be derived directly from the LDETECT signal, see [Figure 21: GPIO Port and the GPIO pin details](#) on page 112. [Figure 22: DETECT signal behavior](#) on page 113 illustrates the DETECT signals behaviour for these two alternatives.



**Figure 22: DETECT signal behavior**

The input buffer of a GPIO pin can be disconnected from the pin to enable power savings when the pin is not used as an input, see [Figure 21: GPIO Port and the GPIO pin details](#) on page 112. Inputs must be connected in order to get a valid input value in the IN register and for the sense mechanism to get access to the pin.

Other peripherals in the system can attach themselves to GPIO pins and override their output value and configuration, or read their analog or digital input value, see [Figure 21: GPIO Port and the GPIO pin details](#) on page 112.

Selected pins also support analog input signals, see ANAIN in [Figure 21: GPIO Port and the GPIO pin details](#) on page 112. The assignment of the analog pins can be found in [Pin assignments](#) on page 13.

**Important:** When a pin is configured as digital input, care has been taken in the nRF52832 design to minimize increased current consumption when the input voltage is between  $V_{IL}$  and  $V_{IH}$ . However, it is a good practice to ensure that the external circuitry does not drive that pin to levels between  $V_{IL}$  and  $V_{IH}$  for a long period of time.

## 20.2 GPIO located near the RADIO

Radio performance parameters, such as sensitivity, may be affected by high frequency digital I/O with large sink/source current close to the radio power supply and antenna pins.

Refer to [Pin assignments](#) on page 13 for recommended usage guidelines to maximize radio performance in an application.

## 20.3 Registers

**Table 28: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x50000000	GPIO	GPIO	General purpose input and output	Deprecated
0x50000000	GPIO	P0	General purpose input and output	

**Table 29: Register Overview**

Register	Offset	Description
<i>OUT</i>	0x504	Write GPIO port
<i>OUTSET</i>	0x508	Set individual bits in GPIO port
<i>OUTCLR</i>	0x50C	Clear individual bits in GPIO port
<i>IN</i>	0x510	Read GPIO port
<i>DIR</i>	0x514	Direction of GPIO pins
<i>DIRSET</i>	0x518	DIR set register
<i>DIRCLR</i>	0x51C	DIR clear register
<i>LATCH</i>	0x520	Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers
<i>DETECTMODE</i>	0x524	Select between default DETECT signal behaviour and LDETECT mode
<i>PIN_CNF[0]</i>	0x700	Configuration of GPIO pins
<i>PIN_CNF[1]</i>	0x704	Configuration of GPIO pins
<i>PIN_CNF[2]</i>	0x708	Configuration of GPIO pins
<i>PIN_CNF[3]</i>	0x70C	Configuration of GPIO pins
<i>PIN_CNF[4]</i>	0x710	Configuration of GPIO pins
<i>PIN_CNF[5]</i>	0x714	Configuration of GPIO pins
<i>PIN_CNF[6]</i>	0x718	Configuration of GPIO pins
<i>PIN_CNF[7]</i>	0x71C	Configuration of GPIO pins
<i>PIN_CNF[8]</i>	0x720	Configuration of GPIO pins
<i>PIN_CNF[9]</i>	0x724	Configuration of GPIO pins
<i>PIN_CNF[10]</i>	0x728	Configuration of GPIO pins
<i>PIN_CNF[11]</i>	0x72C	Configuration of GPIO pins
<i>PIN_CNF[12]</i>	0x730	Configuration of GPIO pins
<i>PIN_CNF[13]</i>	0x734	Configuration of GPIO pins
<i>PIN_CNF[14]</i>	0x738	Configuration of GPIO pins
<i>PIN_CNF[15]</i>	0x73C	Configuration of GPIO pins
<i>PIN_CNF[16]</i>	0x740	Configuration of GPIO pins
<i>PIN_CNF[17]</i>	0x744	Configuration of GPIO pins
<i>PIN_CNF[18]</i>	0x748	Configuration of GPIO pins
<i>PIN_CNF[19]</i>	0x74C	Configuration of GPIO pins
<i>PIN_CNF[20]</i>	0x750	Configuration of GPIO pins
<i>PIN_CNF[21]</i>	0x754	Configuration of GPIO pins
<i>PIN_CNF[22]</i>	0x758	Configuration of GPIO pins
<i>PIN_CNF[23]</i>	0x75C	Configuration of GPIO pins
<i>PIN_CNF[24]</i>	0x760	Configuration of GPIO pins
<i>PIN_CNF[25]</i>	0x764	Configuration of GPIO pins
<i>PIN_CNF[26]</i>	0x768	Configuration of GPIO pins
<i>PIN_CNF[27]</i>	0x76C	Configuration of GPIO pins
<i>PIN_CNF[28]</i>	0x770	Configuration of GPIO pins
<i>PIN_CNF[29]</i>	0x774	Configuration of GPIO pins
<i>PIN_CNF[30]</i>	0x778	Configuration of GPIO pins
<i>PIN_CNF[31]</i>	0x77C	Configuration of GPIO pins

### 20.3.1 OUT

Address offset: 0x504

Write GPIO port

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																													
Id	f	e	d	c	b	a	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a																													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																														
Id	RW	Field	Value																																																										
A	RW	PIN0	Value																																																										

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			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			
Id	RW	Field	Value	Id	Value	Description																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
C	RW	PIN2				Pin 2																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
D	RW	PIN3				Pin 3																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
E	RW	PIN4				Pin 4																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
F	RW	PIN5				Pin 5																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
G	RW	PIN6				Pin 6																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
H	RW	PIN7				Pin 7																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
I	RW	PIN8				Pin 8																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
J	RW	PIN9				Pin 9																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
K	RW	PIN10				Pin 10																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
L	RW	PIN11				Pin 11																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
M	RW	PIN12				Pin 12																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
N	RW	PIN13				Pin 13																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
O	RW	PIN14				Pin 14																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
P	RW	PIN15				Pin 15																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
Q	RW	PIN16				Pin 16																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
R	RW	PIN17				Pin 17																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
S	RW	PIN18				Pin 18																												
		Low	0			Pin driver is low																												
		High	1			Pin driver is high																												
T	RW	PIN19				Pin 19																												
		Low	0			Pin driver is low																												

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
			High	1	Pin driver is high																											
U	RW	PIN20			Pin 20																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
V	RW	PIN21			Pin 21																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
W	RW	PIN22			Pin 22																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
X	RW	PIN23			Pin 23																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
Y	RW	PIN24			Pin 24																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
Z	RW	PIN25			Pin 25																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
a	RW	PIN26			Pin 26																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
b	RW	PIN27			Pin 27																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
c	RW	PIN28			Pin 28																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
d	RW	PIN29			Pin 29																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
e	RW	PIN30			Pin 30																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											
f	RW	PIN31			Pin 31																											
			Low	0	Pin driver is low																											
			High	1	Pin driver is high																											

### **20.3.2 OUTSET**

Address offset: 0x508

Set individual bits in GPIO port

Read: reads value of OUT register.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value		Description																										
A	RW	PIN0				Pin 0																										
		Low		0		Read: pin driver is low																										
		High		1		Read: pin driver is high																										
		Set		1		Write: writing a '1' sets the pin high; writing a '0' has no effect																										
B	RW	PIN1				Pin 1																										
		Low		0		Read: pin driver is low																										

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
C	RW	PIN2	High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
D	RW	PIN3	Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
E	RW	PIN4	Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
F	RW	PIN5	High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
G	RW	PIN6	Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
H	RW	PIN7	Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
I	RW	PIN8	High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
J	RW	PIN9	Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
K	RW	PIN10	Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
L	RW	PIN11	High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
M	RW	PIN12	Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
N	RW	PIN13	Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
O	RW	PIN14	High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
P	RW	PIN15	Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Low	0	Read: pin driver is low																											
			High	1	Read: pin driver is high																											
			Set	1	Write: writing a '1' sets the pin high; writing a '0' has no effect																											
			Pin 15																													

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
Q	RW	PIN16			Pin 16																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
R	RW	PIN17			Pin 17																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
S	RW	PIN18			Pin 18																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
T	RW	PIN19			Pin 19																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
U	RW	PIN20			Pin 20																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
V	RW	PIN21			Pin 21																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
W	RW	PIN22			Pin 22																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
X	RW	PIN23			Pin 23																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
Y	RW	PIN24			Pin 24																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
Z	RW	PIN25			Pin 25																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
a	RW	PIN26			Pin 26																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
b	RW	PIN27			Pin 27																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											
c	RW	PIN28			Pin 28																											
		Low	0		Read: pin driver is low																											
		High	1		Read: pin driver is high																											
		Set	1		Write: writing a '1' sets the pin high; writing a '0' has no effect																											

### 20.3.3 OUTCLR

Address offset: 0x50C

Clear individual bits in GPIO port

Read: reads value of OUT register.



20.3.4 IN

Address offset: 0x510

## Read GPIO port



### **20.3.5 DIR**

Address offset: 0x514

## Direction of GPIO pins

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
A	RW	PIN0				Pin 0																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
B	RW	PIN1				Pin 1																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
C	RW	PIN2				Pin 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		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		
Id	RW	Field	Value	Id	Value	Description																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
D	RW	PIN3				Pin 3																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
E	RW	PIN4				Pin 4																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
F	RW	PIN5				Pin 5																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
G	RW	PIN6				Pin 6																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
H	RW	PIN7				Pin 7																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
I	RW	PIN8				Pin 8																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
J	RW	PIN9				Pin 9																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
K	RW	PIN10				Pin 10																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
L	RW	PIN11				Pin 11																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
M	RW	PIN12				Pin 12																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
N	RW	PIN13				Pin 13																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
O	RW	PIN14				Pin 14																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
P	RW	PIN15				Pin 15																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
Q	RW	PIN16				Pin 16																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
R	RW	PIN17				Pin 17																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
S	RW	PIN18				Pin 18																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
T	RW	PIN19				Pin 19																											
			Input		0	Pin set as input																											
			Output		1	Pin set as output																											
U	RW	PIN20				Pin 20																											
			Input		0	Pin set as input																											

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
			Output	1		Pin set as output																										
V	RW	PIN21		0		Pin 21																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
W	RW	PIN22		0		Pin 22																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
X	RW	PIN23		0		Pin 23																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
Y	RW	PIN24		0		Pin 24																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
Z	RW	PIN25		0		Pin 25																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
a	RW	PIN26		0		Pin 26																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
b	RW	PIN27		0		Pin 27																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
c	RW	PIN28		0		Pin 28																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
d	RW	PIN29		0		Pin 29																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
e	RW	PIN30		0		Pin 30																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										
f	RW	PIN31		0		Pin 31																										
			Input	0		Pin set as input																										
			Output	1		Pin set as output																										

### **20.3.6 DIRSET**

Address offset: 0x518

DIR set register

Read: reads value of DIR register.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
Id	RW	Field	Value Id	Value		Description																									
D	RW	PIN3	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 3																									
E	RW	PIN4	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 4																									
F	RW	PIN5	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 5																									
G	RW	PIN6	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 6																									
H	RW	PIN7	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 7																									
I	RW	PIN8	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 8																									
J	RW	PIN9	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 9																									
K	RW	PIN10	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 10																									
L	RW	PIN11	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 11																									
M	RW	PIN12	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 12																									
N	RW	PIN13	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 13																									
O	RW	PIN14	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 14																									
P	RW	PIN15	Input	0		Read: pin set as input																									
			Output	1		Read: pin set as output																									
			Set	1		Write: writing a '1' sets pin to output; writing a '0' has no effect																									
						Set as output pin 15																									

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																										
Q	RW	PIN16			Set as output pin 16																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
R	RW	PIN17			Set as output pin 17																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
S	RW	PIN18			Set as output pin 18																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
T	RW	PIN19			Set as output pin 19																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
U	RW	PIN20			Set as output pin 20																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
V	RW	PIN21			Set as output pin 21																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
W	RW	PIN22			Set as output pin 22																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
X	RW	PIN23			Set as output pin 23																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
Y	RW	PIN24			Set as output pin 24																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
Z	RW	PIN25			Set as output pin 25																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
a	RW	PIN26			Set as output pin 26																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
b	RW	PIN27			Set as output pin 27																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
c	RW	PIN28			Set as output pin 28																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
			Set	1	Write: writing a '1' sets pin to output; writing a '0' has no effect																										
d	RW	PIN29			Set as output pin 29																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										

### **20.3.7 DIRCLR**

Address offset: 0x51C

## DIR clear register

Read: reads value of DIR register.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																										
J	RW	PIN9	Output	1	Read: pin set as output																										
			Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Read: pin set as input																										
			Output	1	Read: pin set as output																										
K	RW	PIN10	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 10																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
L	RW	PIN11	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 11																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
M	RW	PIN12	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 12																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
N	RW	PIN13	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 13																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
O	RW	PIN14	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 14																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
P	RW	PIN15	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 15																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
Q	RW	PIN16	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 16																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
R	RW	PIN17	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 17																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
S	RW	PIN18	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 18																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
T	RW	PIN19	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 19																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
U	RW	PIN20	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 20																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
V	RW	PIN21	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
			Input	0	Set as input pin 21																										
			Output	1	Read: pin set as input																										
			Clear	1	Read: pin set as output																										
W	RW	PIN22	Clear	1	Write: writing a '1' sets pin to input; writing a '0' has no effect																										
					Set as input pin 22																										

### **20.3.8 LATCH**

Address offset: 0x520

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

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	PINO			Status on whether PIN0 has met criteria set in PIN_CNFO.SENSE register. Write '1' to clear.																											
		NotLatched	0		Criteria has not been met																											
		Latched	1		Criteria has been met																											
B	RW	PIN1			Status on whether PIN1 has met criteria set in PIN_CNF1.SENSE register. Write '1' to clear.																											

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		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
<b>Reset Ox00000000</b>		<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
Id	RW	Field	Value	Id	Value	Description																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
C	RW	PIN2				Status on whether PIN2 has met criteria set in PIN_CNF2.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
D	RW	PIN3				Status on whether PIN3 has met criteria set in PIN_CNF3.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
E	RW	PIN4				Status on whether PIN4 has met criteria set in PIN_CNF4.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
F	RW	PIN5				Status on whether PIN5 has met criteria set in PIN_CNF5.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
G	RW	PIN6				Status on whether PIN6 has met criteria set in PIN_CNF6.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
H	RW	PIN7				Status on whether PIN7 has met criteria set in PIN_CNF7.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
I	RW	PIN8				Status on whether PIN8 has met criteria set in PIN_CNF8.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
J	RW	PIN9				Status on whether PIN9 has met criteria set in PIN_CNF9.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
K	RW	PIN10				Status on whether PIN10 has met criteria set in PIN_CNF10.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
L	RW	PIN11				Status on whether PIN11 has met criteria set in PIN_CNF11.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
M	RW	PIN12				Status on whether PIN12 has met criteria set in PIN_CNF12.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
N	RW	PIN13				Status on whether PIN13 has met criteria set in PIN_CNF13.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											
O	RW	PIN14				Status on whether PIN14 has met criteria set in PIN_CNF14.SENSE register. Write '1' to clear.																											
				NotLatched	0	Criteria has not been met																											
				Latched	1	Criteria has been met																											

				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
				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	
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																														
P	RW	PIN15				Status on whether PIN15 has met criteria set in PIN_CNF15.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
Q	RW	PIN16				Status on whether PIN16 has met criteria set in PIN_CNF16.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
R	RW	PIN17				Status on whether PIN17 has met criteria set in PIN_CNF17.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
S	RW	PIN18				Status on whether PIN18 has met criteria set in PIN_CNF18.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
T	RW	PIN19				Status on whether PIN19 has met criteria set in PIN_CNF19.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
U	RW	PIN20				Status on whether PIN20 has met criteria set in PIN_CNF20.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
V	RW	PIN21				Status on whether PIN21 has met criteria set in PIN_CNF21.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
W	RW	PIN22				Status on whether PIN22 has met criteria set in PIN_CNF22.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
X	RW	PIN23				Status on whether PIN23 has met criteria set in PIN_CNF23.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
Y	RW	PIN24				Status on whether PIN24 has met criteria set in PIN_CNF24.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
Z	RW	PIN25				Status on whether PIN25 has met criteria set in PIN_CNF25.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
a	RW	PIN26				Status on whether PIN26 has met criteria set in PIN_CNF26.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
b	RW	PIN27				Status on whether PIN27 has met criteria set in PIN_CNF27.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														
			Latched	1		Criteria has been met																														
c	RW	PIN28				Status on whether PIN28 has met criteria set in PIN_CNF28.SENSE register. Write '1' to clear.																														
			NotLatched	0		Criteria has not been met																														

### 20.3.9 DETECTMODE

Address offset: 0x524

Select between default DETECT signal behaviour and LDETECT mode

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

### **20.3.10 PIN\_CNF[0]**

Address offset: 0x700

## Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.11 PIN\_CNF[1]

Address offset: 0x704

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.12 PIN\_CNF[2]

Address offset: 0x708

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.13 PIN\_CNF[3]

Address offset: 0x70C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.14 PIN\_CNF[4]

Address offset: 0x710

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.15 PIN\_CNF[5]

Address offset: 0x714

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.16 PIN\_CNF[6]

Address offset: 0x718

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.17 PIN\_CNF[7]

Address offset: 0x71C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.18 PIN\_CNF[8]

Address offset: 0x720

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.19 PIN\_CNF[9]

Address offset: 0x724

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.20 PIN\_CNF[10]

Address offset: 0x728

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.21 PIN\_CNF[11]

Address offset: 0x72C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.22 PIN\_CNF[12]

Address offset: 0x730

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.23 PIN\_CNF[13]

Address offset: 0x734

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.24 PIN\_CNF[14]

Address offset: 0x738

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.25 PIN\_CNF[15]

Address offset: 0x73C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.26 PIN\_CNF[16]

Address offset: 0x740

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.27 PIN\_CNF[17]

Address offset: 0x744

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.28 PIN\_CNF[18]

Address offset: 0x748

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.29 PIN\_CNF[19]

Address offset: 0x74C

Configuration of GPIO pins

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

### 20.3.30 PIN\_CNF[20]

Address offset: 0x750

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.31 PIN\_CNF[21]

Address offset: 0x754

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E RW SENSE				Pin sensing mechanism	
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.32 PIN\_CNF[22]

Address offset: 0x758

Configuration of GPIO pins

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

### 20.3.33 PIN\_CNF[23]

Address offset: 0x75C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.34 PIN\_CNF[24]

Address offset: 0x760

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.35 PIN\_CNF[25]

Address offset: 0x764

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0	Configure pin as an input pin	
		Output	1	Configure pin as an output pin	
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0	Connect input buffer	
		Disconnect	1	Disconnect input buffer	
C	RW	PULL			Pull configuration
		Disabled	0	No pull	
		Pulldown	1	Pull down on pin	
		Pullup	3	Pull up on pin	
D	RW	DRIVE			Drive configuration
		S0S1	0	Standard '0', standard '1'	
		H0S1	1	High drive '0', standard '1'	
		SOH1	2	Standard '0', high drive '1'	
		H0H1	3	High drive '0', high 'drive '1''	
		D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)	
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.36 PIN\_CNF[26]

Address offset: 0x768

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.37 PIN\_CNF[27]

Address offset: 0x76C

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.38 PIN\_CNF[28]

Address offset: 0x770

Configuration of GPIO pins

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

### 20.3.39 PIN\_CNF[29]

Address offset: 0x774

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)
		DOH1	5		Disconnect '0', high drive '1' (normally used for wired-or connections)
		SOD1	6		Standard '0'. disconnect '1' (normally used for wired-and connections)
		HOD1	7		High drive '0', disconnect '1' (normally used for wired-and connections)
E	RW	SENSE			Pin sensing mechanism
		Disabled	0		Disabled
		High	2		Sense for high level
		Low	3		Sense for low level

### 20.3.40 PIN\_CNF[30]

Address offset: 0x778

Configuration of GPIO pins

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
<b>Reset 0x00000002</b>	<b>0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register
		Input	0		Configure pin as an input pin
		Output	1		Configure pin as an output pin
B	RW	INPUT			Connect or disconnect input buffer
		Connect	0		Connect input buffer
		Disconnect	1		Disconnect input buffer
C	RW	PULL			Pull configuration
		Disabled	0		No pull
		Pulldown	1		Pull down on pin
		Pullup	3		Pull up on pin
D	RW	DRIVE			Drive configuration
		SOS1	0		Standard '0', standard '1'
		HOS1	1		High drive '0', standard '1'
		SOH1	2		Standard '0', high drive '1'
		HOH1	3		High drive '0', high 'drive '1''
		DOS1	4		Disconnect '0' standard '1' (normally used for wired-or connections)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E D D D C C B A				
Reset 0x00000002	0 1 0				
Id	RW	Field	Value Id	Value	Description
		D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)	
		S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)	
		H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)	
E	RW	SENSE			Pin sensing mechanism
		Disabled	0	Disabled	
		High	2	Sense for high level	
		Low	3	Sense for low level	

### 20.3.41 PIN\_CNF[31]

Address offset: 0x77C

Configuration of GPIO pins

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

## 20.4 Electrical specification

### 20.4.1 GPIO Electrical Specification

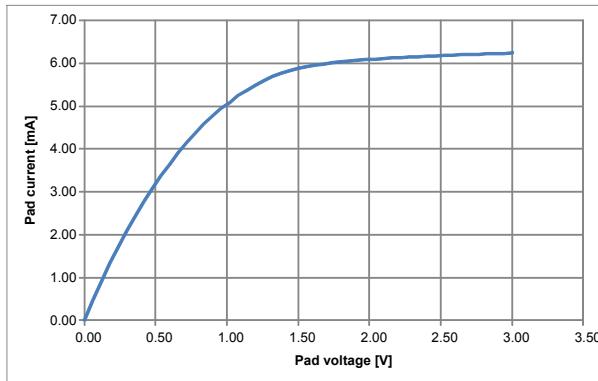
Symbol	Description	Min.	Typ.	Max.	Units
V <sub>IH</sub>	Input high voltage	0.7 x VDD	VDD	V	

Symbol	Description	Min.	Typ.	Max.	Units
V <sub>IL</sub>	Input low voltage	VSS		0.3 x VDD	V
V <sub>OH,SD</sub>	Output high voltage, standard drive, 0.5 mA, VDD $\geq$ 1.7	VDD-0.4		VDD	V
V <sub>OH,HDH</sub>	Output high voltage, high drive, 5 mA, VDD $\geq$ 2.7 V	VDD-0.4		VDD	V
V <sub>OH,HDL</sub>	Output high voltage, high drive, 3 mA, VDD $\geq$ 1.7 V	VDD-0.4		VDD	V
V <sub>OL,SD</sub>	Output low voltage, standard drive, 0.5 mA, VDD $\geq$ 1.7	VSS		VSS+0.4	V
V <sub>OL,HDH</sub>	Output low voltage, high drive, 5 mA, VDD $\geq$ 2.7 V	VSS		VSS+0.4	V
V <sub>OL,HDL</sub>	Output low voltage, high drive, 3 mA, VDD $\geq$ 1.7 V	VSS		VSS+0.4	V
I <sub>OL,SD</sub>	Current at VSS+0.4 V, output set low, standard drive, VDD $\geq$ 1.7	1	2	4	mA
I <sub>OL,HDH</sub>	Current at VSS+0.4 V, output set low, high drive, VDD $\geq$ 2.7 V	6	10	15	mA
I <sub>OL,HDL</sub>	Current at VSS+0.4 V, output set low, high drive, VDD $\geq$ 1.7 V	3			mA
I <sub>OH,SD</sub>	Current at VDD-0.4 V, output set high, standard drive, VDD $\geq$ 1.7	1	2	4	mA
I <sub>OH,HDH</sub>	Current at VDD-0.4 V, output set high, high drive, VDD $\geq$ 2.7 V	6	9	14	mA
I <sub>OH,HDL</sub>	Current at VDD-0.4 V, output set high, high drive, VDD $\geq$ 1.7 V	3			mA
t <sub>RF,15pF</sub>	Rise/fall time, low drive mode, 10-90%, 15 pF load <sup>1</sup>		9		ns
t <sub>RF,25pF</sub>	Rise/fall time, low drive mode, 10-90%, 25 pF load <sup>1</sup>		13		ns
t <sub>RF,50pF</sub>	Rise/fall time, low drive mode, 10-90%, 50 pF load <sup>1</sup>		25		ns
t <sub>HRF,15pF</sub>	Rise/Fall time, high drive mode, 10-90%, 15 pF load <sup>1</sup>		4		ns
t <sub>HRF,25pF</sub>	Rise/Fall time, high drive mode, 10-90%, 25 pF load <sup>1</sup>		5		ns
t <sub>HRF,50pF</sub>	Rise/Fall time, high drive mode, 10-90%, 50 pF load <sup>1</sup>		8		ns
R <sub>PU</sub>	Pull-up resistance	11	13	16	kΩ
R <sub>PD</sub>	Pull-down resistance	11	13	16	kΩ
C <sub>PAD</sub>	Pad capacitance		3		pF
C <sub>PAD_NFC</sub>	Pad capacitance on NFC pads		4		pF
I <sub>NFC_LEAK</sub>	Leakage current between NFC pads when driven to different states	2	10		μA

The current drawn from the battery when GPIO is active as an output is calculated as follows:

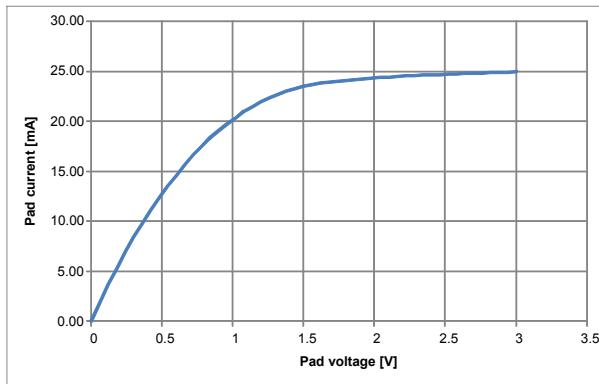
$$I_{GPIO} = V_{DD} C_{load} f$$

C<sub>load</sub> being the load capacitance and "f" is the switching frequency.

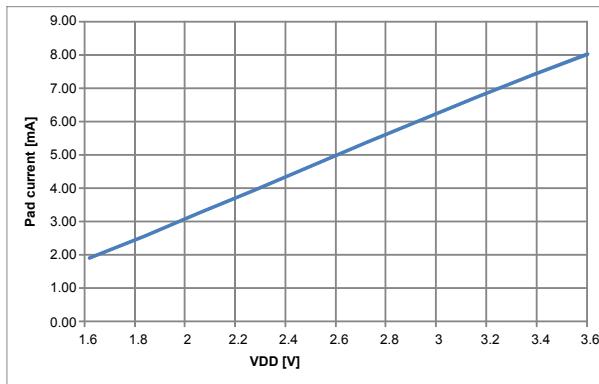


**Figure 23: GPIO drive strength vs Voltage, standard drive, VDD = 3.0 V**

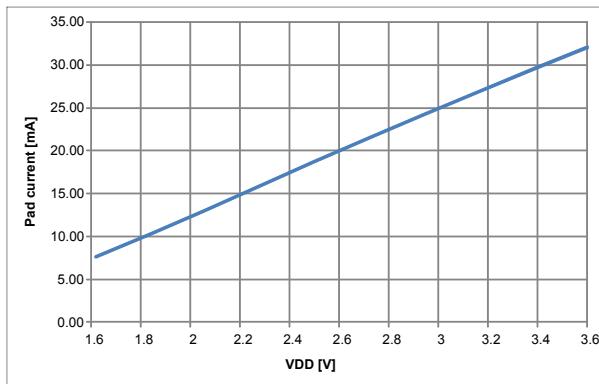
<sup>1</sup> Rise and fall times based on simulations



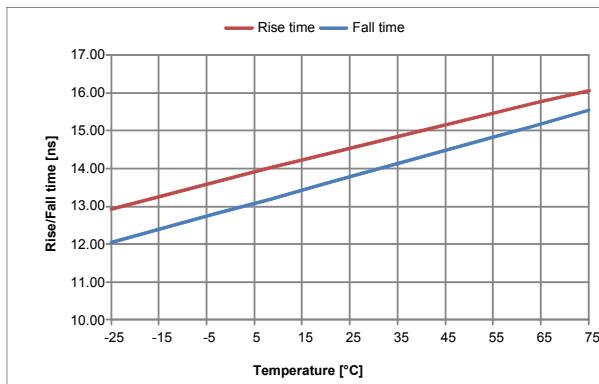
**Figure 24: GPIO drive strength vs Voltage, high drive, VDD = 3.0 V**



**Figure 25: Max sink current vs Voltage, standard drive**



**Figure 26: Max sink current vs Voltage, high drive**



**Figure 27: Rise and fall time vs Temperature, 10%-90%, 25pF load capacitance, VDD = 3.0 V**

## 21 GPIOTE — GPIO tasks and events

The GPIO tasks and events (GPIOTE) module provides functionality for accessing GPIO pins using tasks and events. Each GPIOTE channel can be assigned to one pin.

A GPIOTE block enables GPIOs to generate events on pin state change which can be used to carry out tasks through the PPI system. A GPIO can also be driven to change state on system events using the PPI system. Low power detection of pin state changes is possible when in System ON or System OFF.

**Table 30: GPIOTE properties**

Instance	Number of GPIOTE channels
GPIOTE	8

Up to three tasks can be used in each GPIOTE channel for performing write operations to a pin. Two tasks are fixed (SET and CLR), and one (OUT) is configurable to perform following operations:

- Set
- Clear
- Toggle

An event can be generated in each GPIOTE channel from one of the following input conditions:

- Rising edge
- Falling edge
- Any change

### 21.1 Pin events and tasks

The GPIOTE module has a number of tasks and events that can be configured to operate on individual GPIO pins.

The tasks (SET[n], CLR[n] and OUT[n]) can be used for writing to individual pins, and the events (IN[n]) can be generated from changes occurring at the inputs of individual pins.

The SET task will set the pin selected in CONFIG[n].PSEL to high.

The CLR task will set the pin low.

The effect of the OUT task on the pin is configurable in CONFIG[n].POLARITY , and can either set the pin high, set it low, or toggle it.

The tasks and events are configured using the CONFIG[n] registers. Every set of SET, CLR and OUT[n] tasks and IN[n] events has one CONFIG[n] register associated with it.

As long as a SET[n], CLR[n] and OUT[n] task or an IN[n] event is configured to control a pin **n**, the pin's output value will only be updated by the GPIOTE module. The pin's output value as specified in the GPIO will therefore be ignored as long as the pin is controlled by GPIOTE. Attempting to write a pin as a normal GPIO pin will have no effect. When the GPIOTE is disconnected from a pin, see MODE field in CONFIG[n] register, the associated pin will get the output and configuration values specified in the GPIO module.

When conflicting tasks are triggered simultaneously (i.e. during the same clock cycle) in one channel, the precedence of the tasks will be as described in [Table 31: Task priorities](#) on page 157.

**Table 31: Task priorities**

Priority	Task
1	OUT
2	CLR
3	SET

When setting the CONFIG[n] registers, MODE=Disabled does not have the same effect as MODE=Task and POLARITY=None. In the latter case, a CLR or SET task occurring at the exact same time as OUT will end up with no change on the pin, according to the priorities described in the table above.

When a GPIOTE channel is configured to operate on a pin as a task, the initial value of that pin is configured in the OUTINIT field of CONFIG[n].

## 21.2 Port event

PORTE is an event that can be generated from multiple input pins using the GPIO DETECT signal.

The event will be generated on the rising edge of the DETECT signal. See [GPIO — General purpose input/output](#) on page 111 for more information about the DETECT signal.

Putting the system into System ON IDLE while DETECT is high will not cause DETECT to wake the system up again. Make sure to clear all DETECT sources before entering sleep. If the LATCH register is used as a source, if any bit in LATCH is still high after clearing all or part of the register (for instance due to one of the PINx.DETECT signal still high), a new rising edge will be generated on DETECT, see [Pin configuration](#) on page 111.

Trying to put the system to System OFF while DETECT is high will cause a wakeup from System OFF reset.

This feature is always enabled although the peripheral itself appears to be IDLE, that is, no clocks or other power intensive infrastructure have to be requested to keep this feature enabled. This feature can therefore be used to wake up the CPU from a WFI or WFE type sleep in System ON with all peripherals and the CPU idle, that is, lowest power consumption in System ON mode.

In order to prevent spurious interrupts from the PORT event while configuring the sources, the user shall first disable interrupts on the PORT event (through INTENCLR.PORT), then configure the sources (PIN\_CNF[n].SENSE), clear any potential event that could have occurred during configuration (write '1' to EVENTS\_PORT), and finally enable interrupts (through INTENSET.PORT).

## 21.3 Tasks and events pin configuration

Each GPIOTE channel is associated with one physical GPIO pin through the CONFIG.PSEL field.

When Event mode is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will be configured as an input, overriding the DIR setting in GPIO. Similarly, when Task mode is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will be configured as an output overriding the DIR setting and OUT value in GPIO. When Disabled is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will use its configuration from the PIN[n].CNF registers in GPIO.

Only one GPIOTE channel can be assigned to one physical pin. Failing to do so may result in unpredictable behavior.

## 21.4 Registers

**Table 32: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40006000	GPIOTE	GPIOTE	GPIO Tasks and Events	

**Table 33: Register Overview**

Register	Offset	Description
TASKS_OUT[0]	0x000	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is configured in CONFIG[0].POLARITY.
TASKS_OUT[1]	0x004	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is configured in CONFIG[1].POLARITY.

Register	Offset	Description
TASKS_OUT[2]	0x008	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is configured in CONFIG[2].POLARITY.
TASKS_OUT[3]	0x00C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is configured in CONFIG[3].POLARITY.
TASKS_OUT[4]	0x010	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is configured in CONFIG[4].POLARITY.
TASKS_OUT[5]	0x014	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is configured in CONFIG[5].POLARITY.
TASKS_OUT[6]	0x018	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is configured in CONFIG[6].POLARITY.
TASKS_OUT[7]	0x01C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is configured in CONFIG[7].POLARITY.
TASKS_SET[0]	0x030	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it high.
TASKS_SET[1]	0x034	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it high.
TASKS_SET[2]	0x038	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it high.
TASKS_SET[3]	0x03C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is to set it high.
TASKS_SET[4]	0x040	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is to set it high.
TASKS_SET[5]	0x044	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is to set it high.
TASKS_SET[6]	0x048	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is to set it high.
TASKS_SET[7]	0x04C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is to set it high.
TASKS_CLR[0]	0x060	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it low.
TASKS_CLR[1]	0x064	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it low.
TASKS_CLR[2]	0x068	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it low.
TASKS_CLR[3]	0x06C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is to set it low.
TASKS_CLR[4]	0x070	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is to set it low.
TASKS_CLR[5]	0x074	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is to set it low.
TASKS_CLR[6]	0x078	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is to set it low.
TASKS_CLR[7]	0x07C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is to set it low.
EVENTS_IN[0]	0x100	Event generated from pin specified in CONFIG[0].PSEL
EVENTS_IN[1]	0x104	Event generated from pin specified in CONFIG[1].PSEL
EVENTS_IN[2]	0x108	Event generated from pin specified in CONFIG[2].PSEL
EVENTS_IN[3]	0x10C	Event generated from pin specified in CONFIG[3].PSEL
EVENTS_IN[4]	0x110	Event generated from pin specified in CONFIG[4].PSEL
EVENTS_IN[5]	0x114	Event generated from pin specified in CONFIG[5].PSEL
EVENTS_IN[6]	0x118	Event generated from pin specified in CONFIG[6].PSEL
EVENTS_IN[7]	0x11C	Event generated from pin specified in CONFIG[7].PSEL
EVENTS_PORT	0x17C	Event generated from multiple input GPIO pins with SENSE mechanism enabled
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>CONFIG[0]</b>	0x510	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[1]</b>	0x514	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[2]</b>	0x518	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[3]</b>	0x51C	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[4]</b>	0x520	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[5]</b>	0x524	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[6]</b>	0x528	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event
<b>CONFIG[7]</b>	0x52C	Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

## 21.4.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	H G F E D C B A																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Id	RW	Field	Value																										Description			
A	RW	IN0	Write '1' to Enable interrupt for IN[0] event																													

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																												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			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																														
					See <a href="#">EVENTS_IN[0]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>B</b>	<b>RW</b>	<b>IN1</b>			Write '1' to Enable interrupt for IN[1] event																														
					See <a href="#">EVENTS_IN[1]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>C</b>	<b>RW</b>	<b>IN2</b>			Write '1' to Enable interrupt for IN[2] event																														
					See <a href="#">EVENTS_IN[2]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>D</b>	<b>RW</b>	<b>IN3</b>			Write '1' to Enable interrupt for IN[3] event																														
					See <a href="#">EVENTS_IN[3]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>E</b>	<b>RW</b>	<b>IN4</b>			Write '1' to Enable interrupt for IN[4] event																														
					See <a href="#">EVENTS_IN[4]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>F</b>	<b>RW</b>	<b>IN5</b>			Write '1' to Enable interrupt for IN[5] event																														
					See <a href="#">EVENTS_IN[5]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>G</b>	<b>RW</b>	<b>IN6</b>			Write '1' to Enable interrupt for IN[6] event																														
					See <a href="#">EVENTS_IN[6]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>H</b>	<b>RW</b>	<b>IN7</b>			Write '1' to Enable interrupt for IN[7] event																														
					See <a href="#">EVENTS_IN[7]</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															
<b>I</b>	<b>RW</b>	<b>PORT</b>			Write '1' to Enable interrupt for PORT event																														
					See <a href="#">EVENTS_PORT</a>																														
		Set	1	Enable																															
		Disabled	0	Read: Disabled																															
		Enabled	1	Read: Enabled																															

## 21.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																														
A	RW	IN0			Write '1' to Disable interrupt for IN[0] event																														
					See <a href="#">EVENTS_IN[0]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
B	RW	IN1			Write '1' to Disable interrupt for IN[1] event																														
					See <a href="#">EVENTS_IN[1]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
C	RW	IN2			Write '1' to Disable interrupt for IN[2] event																														
					See <a href="#">EVENTS_IN[2]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
D	RW	IN3			Write '1' to Disable interrupt for IN[3] event																														
					See <a href="#">EVENTS_IN[3]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
E	RW	IN4			Write '1' to Disable interrupt for IN[4] event																														
					See <a href="#">EVENTS_IN[4]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
F	RW	IN5			Write '1' to Disable interrupt for IN[5] event																														
					See <a href="#">EVENTS_IN[5]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
G	RW	IN6			Write '1' to Disable interrupt for IN[6] event																														
					See <a href="#">EVENTS_IN[6]</a>																														
		Clear		1	Disable																														
		Disabled		0	Read: Disabled																														
		Enabled		1	Read: Enabled																														
I	RW	PORT			Write '1' to Disable interrupt for PORT event																														
					See <a href="#">EVENTS_PORT</a>																														
		Clear		1	Disable																														

#### **21.4.4 CONFIG[1]**

Address offset: 0x514

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Id		D	C	C	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A									
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id																											Description					
A RW MODE																													Mode						
																													Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.						
																													Event mode						
																													The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.						
																													Task mode						
																													The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.						

## 21.4.5 CONFIG[2]

Address offset: 0x518

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		D	C	C	B	B	B	B																				A	A			
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>			
Id	RW	Field	Value	Id	Value	Description																										
A	RW	MODE				Mode																										
		Disabled	0			Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.																										
		Event	1			Event mode																										
						The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.																										
		Task	3			Task mode																										
						The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.																										
B	RW	PSEL	[0..31]			GPIO number associated with SET[n], CLR[n] and OUT[n] tasks and IN[n] event																										
C	RW	POLARITY				When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.																										
		None	0			Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.																										
		LoToHi	1			Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.																										
		HiToLo	2			Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.																										
		Toggle	3			Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.																										

Bit number	31	30	29	28	27	26	25	24	23	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	C	B	B	B	B	A	A	
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id		Value		Description																									
D	RW	OUTINIT	When in task mode: Initial value of the output when the GPIOE channel is configured. When in event mode: No effect.																								When in task mode: Initial value of the output when the GPIOE channel is configured. When in event mode: No effect.					
			Task mode: Initial value of pin before task triggering is low																								Task mode: Initial value of pin before task triggering is low					
			Low																								Task mode: Initial value of pin before task triggering is low					
			High																								Task mode: Initial value of pin before task triggering is high					

## 21.4.6 CONFIG[3]

Address offset: 0x51C

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				D	C	C	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																												
A	RW	MODE				Mode																												
		Disabled	0			Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.																												
		Event	1			Event mode																												
						The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.																												
		Task	3			Task mode																												
						The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.																												
B	RW	PSEL	[0..31]			GPIO number associated with SET[n], CLR[n] and OUT[n] tasks and IN[n] event																												
C	RW	POLARITY				When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.																												
		None	0			Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.																												
		LoToHi	1			Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.																												
		HiToLo	2			Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.																												
		Toggle	3			Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.																												
D	RW	OUTINIT				When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.																												
		Low	0			Task mode: Initial value of pin before task triggering is low																												
		High	1			Task mode: Initial value of pin before task triggering is high																												

## 21.4.7 CONFIG[4]

Address offset: 0x520

## Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		D	C	C	B	B	B	B	A	A																						
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																										
A	RW	MODE				Mode																										
		Disabled		0		Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.																										
		Event		1		Event mode																										
						The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.																										
		Task		3		Task mode																										
						The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.																										
B	RW	PSEL		[0..31]		GPIO number associated with SET[n], CLR[n] and OUT[n] tasks and IN[n] event																										
C	RW	POLARITY				When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.																										
		None		0		Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.																										
		LoToHi		1		Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.																										
		HiToLo		2		Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.																										
		Toggle		3		Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.																										
D	RW	OUTINIT				When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.																										
		Low		0		Task mode: Initial value of pin before task triggering is low																										
		High		1		Task mode: Initial value of pin before task triggering is high																										

## 21.4.8 CONFIG[5]

Address offset: 0x524

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Id															D	C	C	B	B	B	B	B	A	A											
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value	Description																														
B	RW	PSEL		[0..31]	GPIO number associated with SET[n], CLR[n] and OUT[n] tasks and IN[n] event																														
C	RW	POLARITY			When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.																														
		None	0		Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.																														
		LoToHi	1		Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.																														
		HiToLo	2		Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.																														
		Toggle	3		Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.																														
D	RW	OUTINIT			When in task mode: Initial value of the output when the GPIOE channel is configured. When in event mode: No effect.																														
		Low	0		Task mode: Initial value of pin before task triggering is low																														
		High	1		Task mode: Initial value of pin before task triggering is high																														

#### **21.4.9 CONFIG[6]**

Address offset: 0x528

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id												D	C	C			B	B	B	B	B	B	A	A									
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id																								Description						
D	RW	OUTINIT																									When in task mode: Initial value of the output when the GPIOE channel is configured. When in event mode: No effect.						
																											Task mode: Initial value of pin before task triggering is low						
																											Task mode: Initial value of pin before task triggering is high						

#### **21.4.10 CONFIG[7]**

Address offset: 0x52C

Configuration for OUT[n], SET[n] and CLR[n] tasks and IN[n] event

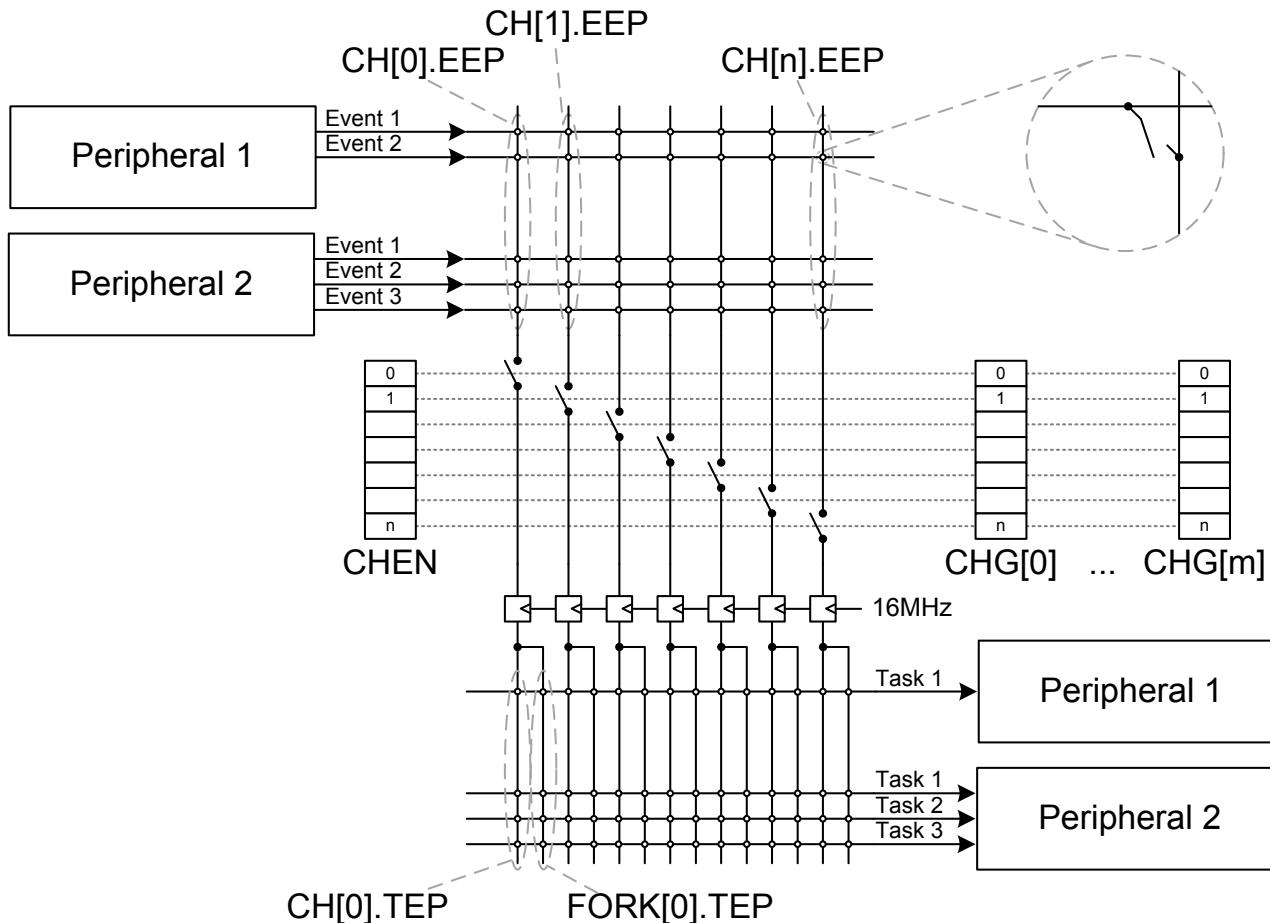
## **21.5 Electrical specification**

### **21.5.1 GPIOTE Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>GPIOE,IN</sub>	Run current with 1 or more GPIOE active channels in Input mode	0.1	0.5	μA	

## 22 PPI — Programmable peripheral interconnect

The Programmable peripheral interconnect (PPI) enables peripherals to interact autonomously with each other using tasks and events independent of the CPU. The PPI allows precise synchronization between peripherals when real-time application constraints exist and eliminates the need for CPU activity to implement behavior which can be predefined using PPI.



**Figure 28: PPI block diagram**

The PPI system has, in addition to the fully programmable peripheral interconnections, a set of channels where the event end point (EEP) and task end points (TEP) are fixed in hardware. These fixed channels can be individually enabled, disabled, or added to PPI channel groups in the same way as ordinary PPI channels.

**Table 34: Configurable and fixed PPI channels**

Instance	Channel	Number of channels	Number of groups
PPI	0-19	20	6
PPI (fixed)	20-31	12	

The PPI provides a mechanism to automatically trigger a task in one peripheral as a result of an event occurring in another peripheral. A task is connected to an event through a PPI channel. The PPI channel is composed of three end point registers, one EEP and two TEPs. A peripheral task is connected to a TEP using the address of the task register associated with the task. Similarly, a peripheral event is connected to an EEP using the address of the event register associated with the event.

On each PPI channel, the signals are synchronized to the 16 MHz clock, to avoid any internal violation of setup and hold timings. As a consequence, events that are synchronous to the 16 MHz clock will be delayed by one clock period, while other asynchronous events will be delayed by up to one 16 MHz clock period.

Each TEP implements a fork mechanism that enables a second task to be triggered at the same time as the task specified in the TEP is triggered. This second task is configured in the task end point register in the FORK registers groups, e.g. FORK.TEP[0] is associated with PPI channel CH[0].

There are two ways of enabling and disabling PPI channels:

- Enable or disable PPI channels individually using the CHEN, CHENSET, and CHENCLR registers.
- Enable or disable PPI channels in PPI channel groups through the groups' ENABLE and DISABLE tasks. Prior to these tasks being triggered, the PPI channel group must be configured to define which PPI channels belongs to which groups.

Note that when a channel belongs to two groups m and n, and CHG[m].EN and CHG[n].DIS occur simultaneously (m and n can be equal or different), EN on that channel has priority.

PPI tasks (for example, CHG[0].EN) can be triggered through the PPI like any other task, which means they can be hooked up to a PPI channel as a TEP. One event can trigger multiple tasks by using multiple channels and one task can be triggered by multiple events in the same way.

## 22.1 Pre-programmed channels

Some of the PPI channels are pre-programmed. These channels cannot be configured by the CPU, but can be added to groups and enabled and disabled like the general purpose PPI channels. The FORK TEP for these channels are still programmable and can be used by the application.

For a list of pre-programmed PPI channels, see the table below.

**Table 35: Pre-programmed channels**

Channel	EPP	TEP
20	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
21	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
22	TIMER0->EVENTS_COMPARE[1]	RADIO->TASKS_DISABLE
23	RADIO->EVENTS_BCMATCH	AAR->TASKS_START
24	RADIO->EVENTS_READY	CCM->TASKS_KSGEN
25	RADIO->EVENTS_ADDRESS	CCM->TASKS_CRYPT
26	RADIO->EVENTS_ADDRESS	TIMER0->TASKS_CAPTURE[1]
27	RADIO->EVENTS_END	TIMER0->TASKS_CAPTURE[2]
28	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
29	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
30	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_CLEAR
31	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_START

## 22.2 Registers

**Table 36: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4001F000	PPI	PPI	Programmable Peripheral Interconnect	

**Table 37: Register Overview**

Register	Offset	Description
TASKS_CHG[0].EN	0x000	Enable channel group 0
TASKS_CHG[0].DIS	0x004	Disable channel group 0
TASKS_CHG[1].EN	0x008	Enable channel group 1
TASKS_CHG[1].DIS	0x00C	Disable channel group 1
TASKS_CHG[2].EN	0x010	Enable channel group 2
TASKS_CHG[2].DIS	0x014	Disable channel group 2
TASKS_CHG[3].EN	0x018	Enable channel group 3
TASKS_CHG[3].DIS	0x01C	Disable channel group 3
TASKS_CHG[4].EN	0x020	Enable channel group 4
TASKS_CHG[4].DIS	0x024	Disable channel group 4
TASKS_CHG[5].EN	0x028	Enable channel group 5
TASKS_CHG[5].DIS	0x02C	Disable channel group 5

Register	Offset	Description
<i>CHEN</i>	0x500	Channel enable register
<i>CHENSET</i>	0x504	Channel enable set register
<i>CHENCLR</i>	0x508	Channel enable clear register
<i>CH[0].EEP</i>	0x510	Channel 0 event end-point
<i>CH[0].TEP</i>	0x514	Channel 0 task end-point
<i>CH[1].EEP</i>	0x518	Channel 1 event end-point
<i>CH[1].TEP</i>	0x51C	Channel 1 task end-point
<i>CH[2].EEP</i>	0x520	Channel 2 event end-point
<i>CH[2].TEP</i>	0x524	Channel 2 task end-point
<i>CH[3].EEP</i>	0x528	Channel 3 event end-point
<i>CH[3].TEP</i>	0x52C	Channel 3 task end-point
<i>CH[4].EEP</i>	0x530	Channel 4 event end-point
<i>CH[4].TEP</i>	0x534	Channel 4 task end-point
<i>CH[5].EEP</i>	0x538	Channel 5 event end-point
<i>CH[5].TEP</i>	0x53C	Channel 5 task end-point
<i>CH[6].EEP</i>	0x540	Channel 6 event end-point
<i>CH[6].TEP</i>	0x544	Channel 6 task end-point
<i>CH[7].EEP</i>	0x548	Channel 7 event end-point
<i>CH[7].TEP</i>	0x54C	Channel 7 task end-point
<i>CH[8].EEP</i>	0x550	Channel 8 event end-point
<i>CH[8].TEP</i>	0x554	Channel 8 task end-point
<i>CH[9].EEP</i>	0x558	Channel 9 event end-point
<i>CH[9].TEP</i>	0x55C	Channel 9 task end-point
<i>CH[10].EEP</i>	0x560	Channel 10 event end-point
<i>CH[10].TEP</i>	0x564	Channel 10 task end-point
<i>CH[11].EEP</i>	0x568	Channel 11 event end-point
<i>CH[11].TEP</i>	0x56C	Channel 11 task end-point
<i>CH[12].EEP</i>	0x570	Channel 12 event end-point
<i>CH[12].TEP</i>	0x574	Channel 12 task end-point
<i>CH[13].EEP</i>	0x578	Channel 13 event end-point
<i>CH[13].TEP</i>	0x57C	Channel 13 task end-point
<i>CH[14].EEP</i>	0x580	Channel 14 event end-point
<i>CH[14].TEP</i>	0x584	Channel 14 task end-point
<i>CH[15].EEP</i>	0x588	Channel 15 event end-point
<i>CH[15].TEP</i>	0x58C	Channel 15 task end-point
<i>CH[16].EEP</i>	0x590	Channel 16 event end-point
<i>CH[16].TEP</i>	0x594	Channel 16 task end-point
<i>CH[17].EEP</i>	0x598	Channel 17 event end-point
<i>CH[17].TEP</i>	0x59C	Channel 17 task end-point
<i>CH[18].EEP</i>	0x5A0	Channel 18 event end-point
<i>CH[18].TEP</i>	0x5A4	Channel 18 task end-point
<i>CH[19].EEP</i>	0x5A8	Channel 19 event end-point
<i>CH[19].TEP</i>	0x5AC	Channel 19 task end-point
<i>CHG[0]</i>	0x800	Channel group 0
<i>CHG[1]</i>	0x804	Channel group 1
<i>CHG[2]</i>	0x808	Channel group 2
<i>CHG[3]</i>	0x80C	Channel group 3
<i>CHG[4]</i>	0x810	Channel group 4
<i>CHG[5]</i>	0x814	Channel group 5
<i>FORK[0].TEP</i>	0x910	Channel 0 task end-point
<i>FORK[1].TEP</i>	0x914	Channel 1 task end-point
<i>FORK[2].TEP</i>	0x918	Channel 2 task end-point
<i>FORK[3].TEP</i>	0x91C	Channel 3 task end-point
<i>FORK[4].TEP</i>	0x920	Channel 4 task end-point
<i>FORK[5].TEP</i>	0x924	Channel 5 task end-point
<i>FORK[6].TEP</i>	0x928	Channel 6 task end-point
<i>FORK[7].TEP</i>	0x92C	Channel 7 task end-point
<i>FORK[8].TEP</i>	0x930	Channel 8 task end-point

Register	Offset	Description
<i>FORK[9].TEP</i>	0x934	Channel 9 task end-point
<i>FORK[10].TEP</i>	0x938	Channel 10 task end-point
<i>FORK[11].TEP</i>	0x93C	Channel 11 task end-point
<i>FORK[12].TEP</i>	0x940	Channel 12 task end-point
<i>FORK[13].TEP</i>	0x944	Channel 13 task end-point
<i>FORK[14].TEP</i>	0x948	Channel 14 task end-point
<i>FORK[15].TEP</i>	0x94C	Channel 15 task end-point
<i>FORK[16].TEP</i>	0x950	Channel 16 task end-point
<i>FORK[17].TEP</i>	0x954	Channel 17 task end-point
<i>FORK[18].TEP</i>	0x958	Channel 18 task end-point
<i>FORK[19].TEP</i>	0x95C	Channel 19 task end-point
<i>FORK[20].TEP</i>	0x960	Channel 20 task end-point
<i>FORK[21].TEP</i>	0x964	Channel 21 task end-point
<i>FORK[22].TEP</i>	0x968	Channel 22 task end-point
<i>FORK[23].TEP</i>	0x96C	Channel 23 task end-point
<i>FORK[24].TEP</i>	0x970	Channel 24 task end-point
<i>FORK[25].TEP</i>	0x974	Channel 25 task end-point
<i>FORK[26].TEP</i>	0x978	Channel 26 task end-point
<i>FORK[27].TEP</i>	0x97C	Channel 27 task end-point
<i>FORK[28].TEP</i>	0x980	Channel 28 task end-point
<i>FORK[29].TEP</i>	0x984	Channel 29 task end-point
<i>FORK[30].TEP</i>	0x988	Channel 30 task end-point
<i>FORK[31].TEP</i>	0x98C	Channel 31 task end-point

## 22.2.1 CHEN

Address offset: 0x500

Channel enable register

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
<b>Reset 0x00000000</b>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																											
A	RW	CH0				Enable or disable channel 0																											
						Disable channel																											
B	RW	CH1				Enable or disable channel 1																											
						Disable channel																											
C	RW	CH2				Enable or disable channel 2																											
						Disable channel																											
D	RW	CH3				Enable or disable channel 3																											
						Disable channel																											
E	RW	CH4				Enable or disable channel 4																											
						Disable channel																											
F	RW	CH5				Enable or disable channel 5																											
						Disable channel																											
G	RW	CH6				Enable or disable channel 6																											
						Disable channel																											
H	RW	CH7				Enable or disable channel 7																											
						Disable channel																											
I	RW	CH8				Enable or disable channel 8																											



Bit number			31	30	29	28	27	26	25	24	23	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	
Id	RW	Field	Value																										Description					
b			Enabled																										Enable channel					
			Disabled																										Enable or disable channel 27					
			Enabled																										Disable channel					
c			Enabled																										Enable channel					
			Disabled																										Enable or disable channel 28					
			Enabled																										Disable channel					
d			Enabled																										Enable channel					
			Disabled																										Disable or disable channel 29					
			Enabled																										Enable channel					
e			Enabled																										Enable or disable channel 30					
			Disabled																										Disable channel					
			Enabled																										Enable channel					
f			Enabled																										Enable or disable channel 31					
			Disabled																										Disable channel					
			Enabled																										Enable channel					

## 22.2.2 CHENSET

Address offset: 0x504

## Channel enable set register

Read: reads value of CH*i* field in CHEN register.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
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		
Id	RW	Field	Value Id																										Description					
A	RW	CH0																										Channel 0 enable set register. Writing '0' has no effect						
																													Read: channel disabled					
																													Read: channel enabled					
																													Write: Enable channel					
B	RW	CH1																											Channel 1 enable set register. Writing '0' has no effect					
																													Read: channel disabled					
																												Read: channel enabled						
																												Write: Enable channel						
C	RW	CH2																											Channel 2 enable set register. Writing '0' has no effect					
																												Read: channel disabled						
																												Read: channel enabled						
																												Write: Enable channel						
D	RW	CH3																											Channel 3 enable set register. Writing '0' has no effect					
																												Read: channel disabled						
																												Read: channel enabled						
																												Write: Enable channel						
E	RW	CH4																											Channel 4 enable set register. Writing '0' has no effect					
																												Read: channel disabled						
																												Read: channel enabled						
																												Write: Enable channel						
F	RW	CH5																											Channel 5 enable set register. Writing '0' has no effect					
																												Read: channel disabled						
																												Read: channel enabled						
																												Write: Enable channel						
G	RW	CH6																										Channel 6 enable set register. Writing '0' has no effect						
																												Read: channel disabled						
																												Read: channel 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	
Id	RW	Field	Value	Id	Value	Description																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
V	RW	CH21				Channel 21 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
W	RW	CH22				Channel 22 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
X	RW	CH23				Channel 23 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
Y	RW	CH24				Channel 24 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
Z	RW	CH25				Channel 25 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
a	RW	CH26				Channel 26 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
b	RW	CH27				Channel 27 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
c	RW	CH28				Channel 28 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
d	RW	CH29				Channel 29 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
e	RW	CH30				Channel 30 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												
f	RW	CH31				Channel 31 enable set register. Writing '0' has no effect																												
		Disabled	0			Read: channel disabled																												
		Enabled	1			Read: channel enabled																												
		Set	1			Write: Enable channel																												

### 22.2.3 CHENCLR

Address offset: 0x508

Channel enable clear register

Read: reads value of CH{i} field in CHEN register.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																										
A	RW	CH0	Disabled	0	Channel 0 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
B	RW	CH1	Disabled	0	Channel 1 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
C	RW	CH2	Disabled	0	Channel 2 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
D	RW	CH3	Disabled	0	Channel 3 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
E	RW	CH4	Disabled	0	Channel 4 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
F	RW	CH5	Disabled	0	Channel 5 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
G	RW	CH6	Disabled	0	Channel 6 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
H	RW	CH7	Disabled	0	Channel 7 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
I	RW	CH8	Disabled	0	Channel 8 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
J	RW	CH9	Disabled	0	Channel 9 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
K	RW	CH10	Disabled	0	Channel 10 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
L	RW	CH11	Disabled	0	Channel 11 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
M	RW	CH12	Disabled	0	Channel 12 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
			Clear	1	Read: channel enabled																										
					Write: disable channel																										
N	RW	CH13	Disabled	0	Channel 13 enable clear register. Writing '0' has no effect																										
			Enabled	1	Read: channel disabled																										
					Read: channel enabled																										



## 22.2.4 CH[0].EEP

Address offset: 0x510

### Channel 0 event end-point

## 22.2.5 CH[0].TEP

Address offset: 0x514

## Channel 0 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	0
Id	RW	Field	Value																								Description							
A	RW	TEP																									Pointer to task register. Accepts only addresses to registers from the Task group.							

## 22.2.6 CH[1].EEP

Address offset: 0x518

### Channel 1 event end-point

## 22.2.7 CH[1].TEP

Address offset: 0x51C

### Channel 1 task end-point

## 22.2.8 CH[2].EEP

Address offset: 0x520

## Channel 2 event end-point

## 22.2.9 CH[2].TEP

Address offset: 0x524

## Channel 2 task end-point

## 22.2.10 CH[3].EEP

Address offset: 0x528

### Channel 3 event end-point

## 22.2.11 CH[3].TEP

Address offset: 0x52C

### Channel 3 task end-point

## 22.2.12 CH[4].EEP

Address offset: 0x530

## Channel 4 event end-point

## 22.2.13 CH[4].TEP

Address offset: 0x534

## Channel 4 task end-point

22.2.14 CH[5].EEP

Address offset: 0x538

## Channel 5 event end-point

## 22.2.15 CH[5].TEP

Address offset: 0x53C

## Channel 5 task end-point

22.2.16 CH[6].EEP

Address offset: 0x540

## Channel 6 event end-point

## 22.2.17 CH[6].TEP

Address offset: 0x544

## Channel 6 task end-point

22.2.18 CH[7].EEP

Address offset: 0x548

## Channel 7 event end-point

22.2.19 CH[7].TEP

Address offset: 0x54C

## Channel 7 task end-point

22.2.20 CH[8].EEP

Address offset: 0x550

## Channel 8 event end-point

22.2.21 CH[8].TEP

Address offset: 0x554

## Channel 8 task end-point

22.2.22 CH[9].EEP

Address offset: 0x558

## Channel 9 event end-point

22.2.23 CH[9].TEP

Address offset: 0x55C

## Channel 9 task end-point

22.2.24 CH[10].EEP

Address offset: 0x560

## Channel 10 event end-point

22.2.25 CH[10].TEP

Address offset: 0x564

## Channel 10 task end-point

22.2.26 CH[11].EEP

Address offset: 0x568

## Channel 11 event end-point

22.2.27 CH[11].TEP

Address offset: 0x56C

## Channel 11 task end-point

22.2.28 CH[12].EEP

Address offset: 0x570

## Channel 12 event end-point

22.2.29 CH[12].TEP

Address offset: 0x574

## Channel 12 task end-point

22.2.30 CH[13].EEP

Address offset: 0x578

## Channel 13 event end-point

22.2.31 CH[13].TEP

Address offset: 0x57C

## Channel 13 task end-point

22.2.32 CH[14].EEP

Address offset: 0x580

## Channel 14 event end-point

22.2.33 CH[14].TEP

Address offset: 0x584

## Channel 14 task end-point

22.2.34 CH[15].EEP

Address offset: 0x588

## Channel 15 event end-point

22.2.35 CH[15].TEP

Address offset: 0x58C

## Channel 15 task end-point

22.2.36 CH[16].EEP

Address offset: 0x590

## Channel 16 event end-point

22.2.37 CH[16].TEP

Address offset: 0x594

## Channel 16 task end-point

22.2.38 CH[17].EEP

Address offset: 0x598

## Channel 17 event end-point

22.2.39 CH[17].TEP

Address offset: 0x59C

Channel 17 task end-point

22.2.40 CH[18].EEP

Address offset: 0x5A0

## Channel 18 event end-point

22.2.41 CH[18].TEP

Address offset: 0x5A4

## Channel 18 task end-point

22.2.42 CH[19].EEP

Address offset: 0x5A8

## Channel 19 event end-point

22.2.43 CH[19].TEP

Address offset: 0x5AC

## Channel 19 task end-point

## 22.2.44 CHG[0]

Address offset: 0x800

## Channel group 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	
Id	RW	Field	Value Id																										Description					
A	RW	CH0																										Include or exclude channel 0						
																												Exclude						
																												Include						
B	RW	CH1																										Include or exclude channel 1						
																												Exclude						
																												Include						
C	RW	CH2																										Include or exclude channel 2						
																												Exclude						
																												Include						
D	RW	CH3																										Include or exclude channel 3						
																												Exclude						
																												Include						
E	RW	CH4																										Include or exclude channel 4						
																												Exclude						
																												Include						
F	RW	CH5																										Include or exclude channel 5						
																												Exclude						
																												Include						
G	RW	CH6																										Include or exclude channel 6						
																												Exclude						
																												Include						
H	RW	CH7																										Include or exclude channel 7						
																												Exclude						
																												Include						
I	RW	CH8																										Include or exclude channel 8						
																												Exclude						

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
			Included	1	Include																											
J	RW	CH9	Excluded	0	Exclude																											
			Included	1	Include																											
K	RW	CH10			Include or exclude channel 10																											
			Excluded	0	Exclude																											
			Included	1	Include																											
L	RW	CH11			Include or exclude channel 11																											
			Excluded	0	Exclude																											
			Included	1	Include																											
M	RW	CH12			Include or exclude channel 12																											
			Excluded	0	Exclude																											
			Included	1	Include																											
N	RW	CH13			Include or exclude channel 13																											
			Excluded	0	Exclude																											
			Included	1	Include																											
O	RW	CH14			Include or exclude channel 14																											
			Excluded	0	Exclude																											
			Included	1	Include																											
P	RW	CH15			Include or exclude channel 15																											
			Excluded	0	Exclude																											
			Included	1	Include																											
Q	RW	CH16			Include or exclude channel 16																											
			Excluded	0	Exclude																											
			Included	1	Include																											
R	RW	CH17			Include or exclude channel 17																											
			Excluded	0	Exclude																											
			Included	1	Include																											
S	RW	CH18			Include or exclude channel 18																											
			Excluded	0	Exclude																											
			Included	1	Include																											
T	RW	CH19			Include or exclude channel 19																											
			Excluded	0	Exclude																											
			Included	1	Include																											
U	RW	CH20			Include or exclude channel 20																											
			Excluded	0	Exclude																											
			Included	1	Include																											
V	RW	CH21			Include or exclude channel 21																											
			Excluded	0	Exclude																											
			Included	1	Include																											
W	RW	CH22			Include or exclude channel 22																											
			Excluded	0	Exclude																											
			Included	1	Include																											
X	RW	CH23			Include or exclude channel 23																											
			Excluded	0	Exclude																											
			Included	1	Include																											
Y	RW	CH24			Include or exclude channel 24																											
			Excluded	0	Exclude																											
			Included	1	Include																											
Z	RW	CH25			Include or exclude channel 25																											
			Excluded	0	Exclude																											
			Included	1	Include																											
a	RW	CH26			Include or exclude channel 26																											
			Excluded	0	Exclude																											
			Included	1	Include																											

## 22.2.45 CHG[1]

Address offset: 0x804

## Channel group 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	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
A	RW	CH0		Excluded	0	Include or exclude channel 0																										
				Included	1	Exclude																										
B	RW	CH1		Excluded	0	Include or exclude channel 1																										
				Included	1	Exclude																										
C	RW	CH2		Excluded	0	Include or exclude channel 2																										
				Included	1	Exclude																										
D	RW	CH3		Excluded	0	Include or exclude channel 3																										
				Included	1	Exclude																										
E	RW	CH4		Excluded	0	Include or exclude channel 4																										
				Included	1	Exclude																										
F	RW	CH5		Excluded	0	Include or exclude channel 5																										
				Included	1	Exclude																										
G	RW	CH6		Excluded	0	Include or exclude channel 6																										
				Included	1	Exclude																										
H	RW	CH7		Excluded	0	Include or exclude channel 7																										
				Included	1	Exclude																										
I	RW	CH8		Excluded	0	Include or exclude channel 8																										
				Included	1	Exclude																										
J	RW	CH9		Excluded	0	Include or exclude channel 9																										
				Included	1	Exclude																										

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
K	RW	CH10				Include or exclude channel 10																										
			Excluded	0		Exclude																										
			Included	1		Include																										
L	RW	CH11				Include or exclude channel 11																										
			Excluded	0		Exclude																										
			Included	1		Include																										
M	RW	CH12				Include or exclude channel 12																										
			Excluded	0		Exclude																										
			Included	1		Include																										
N	RW	CH13				Include or exclude channel 13																										
			Excluded	0		Exclude																										
			Included	1		Include																										
O	RW	CH14				Include or exclude channel 14																										
			Excluded	0		Exclude																										
			Included	1		Include																										
P	RW	CH15				Include or exclude channel 15																										
			Excluded	0		Exclude																										
			Included	1		Include																										
Q	RW	CH16				Include or exclude channel 16																										
			Excluded	0		Exclude																										
			Included	1		Include																										
R	RW	CH17				Include or exclude channel 17																										
			Excluded	0		Exclude																										
			Included	1		Include																										
S	RW	CH18				Include or exclude channel 18																										
			Excluded	0		Exclude																										
			Included	1		Include																										
T	RW	CH19				Include or exclude channel 19																										
			Excluded	0		Exclude																										
			Included	1		Include																										
U	RW	CH20				Include or exclude channel 20																										
			Excluded	0		Exclude																										
			Included	1		Include																										
V	RW	CH21				Include or exclude channel 21																										
			Excluded	0		Exclude																										
			Included	1		Include																										
W	RW	CH22				Include or exclude channel 22																										
			Excluded	0		Exclude																										
			Included	1		Include																										
X	RW	CH23				Include or exclude channel 23																										
			Excluded	0		Exclude																										
			Included	1		Include																										
Y	RW	CH24				Include or exclude channel 24																										
			Excluded	0		Exclude																										
			Included	1		Include																										
Z	RW	CH25				Include or exclude channel 25																										
			Excluded	0		Exclude																										
			Included	1		Include																										
a	RW	CH26				Include or exclude channel 26																										
			Excluded	0		Exclude																										
			Included	1		Include																										
b	RW	CH27				Include or exclude channel 27																										
			Excluded	0		Exclude																										
			Included	1		Include																										
c	RW	CH28				Include or exclude channel 28																										

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value		Description																										
			Excluded	0		Exclude																										
			Included	1		Include																										
d	RW	CH29		Include or exclude channel 29																												
			Excluded	0		Exclude																										
			Included	1		Include																										
e	RW	CH30		Include or exclude channel 30																												
			Excluded	0		Exclude																										
			Included	1		Include																										
f	RW	CH31		Include or exclude channel 31																												
			Excluded	0		Exclude																										
			Included	1		Include																										

## 22.2.46 CHG[2]

Address offset: 0x808

## Channel group 2

Bit number																								
Id																								
<b>Reset 0x00000000</b>																								
Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	
M	RW	CH12																						
N	RW	CH13																						
O	RW	CH14																						
P	RW	CH15																						
Q	RW	CH16																						
R	RW	CH17																						
S	RW	CH18																						
T	RW	CH19																						
U	RW	CH20																						
V	RW	CH21																						
W	RW	CH22																						
X	RW	CH23																						
Y	RW	CH24																						
Z	RW	CH25																						
a	RW	CH26																						
b	RW	CH27																						
c	RW	CH28																						
d	RW	CH29																						

Bit number	31	30	29	28	27	26	25	24	23	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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
			Included	1	Include																											
e	RW	CH30			Include or exclude channel 30																											
			Excluded	0	Exclude																											
			Included	1	Include																											
f	RW	CH31			Include or exclude channel 31																											
			Excluded	0	Exclude																											
			Included	1	Include																											

22.2.47 CHG[3]

Address offset: 0x80C

### Channel group 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	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
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	CH0			Include or exclude channel 0																											
			Excluded	0	Exclude																											
			Included	1	Include																											
B	RW	CH1			Include or exclude channel 1																											
			Excluded	0	Exclude																											
			Included	1	Include																											
C	RW	CH2			Include or exclude channel 2																											
			Excluded	0	Exclude																											
			Included	1	Include																											
D	RW	CH3			Include or exclude channel 3																											
			Excluded	0	Exclude																											
			Included	1	Include																											
E	RW	CH4			Include or exclude channel 4																											
			Excluded	0	Exclude																											
			Included	1	Include																											
F	RW	CH5			Include or exclude channel 5																											
			Excluded	0	Exclude																											
			Included	1	Include																											
G	RW	CH6			Include or exclude channel 6																											
			Excluded	0	Exclude																											
			Included	1	Include																											
H	RW	CH7			Include or exclude channel 7																											
			Excluded	0	Exclude																											
			Included	1	Include																											
I	RW	CH8			Include or exclude channel 8																											
			Excluded	0	Exclude																											
			Included	1	Include																											
J	RW	CH9			Include or exclude channel 9																											
			Excluded	0	Exclude																											
			Included	1	Include																											
K	RW	CH10			Include or exclude channel 10																											
			Excluded	0	Exclude																											
			Included	1	Include																											
L	RW	CH11			Include or exclude channel 11																											
			Excluded	0	Exclude																											
			Included	1	Include																											
M	RW	CH12			Include or exclude channel 12																											
			Excluded	0	Exclude																											



22.2.48 CHG[4]

Address offset: 0x810

## Channel group 4

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			f	e	d	c	b	a	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		
<b>Id</b>	<b>RW</b>		<b>Field</b>																															
O	RW		CH14																											Include or exclude channel 14				
			Excluded																											Exclude				
			Included																										Include					
P	RW		CH15																										Include or exclude channel 15					
			Excluded																										Exclude					
			Included																									Include						
Q	RW		CH16																										Include or exclude channel 16					
			Excluded																									Exclude						
			Included																									Include						
R	RW		CH17																										Include or exclude channel 17					
			Excluded																									Exclude						
			Included																									Include						
S	RW		CH18																									Include or exclude channel 18						
			Excluded																									Exclude						
			Included																									Include						
T	RW		CH19																									Include or exclude channel 19						
			Excluded																									Exclude						
			Included																									Include						
U	RW		CH20																									Include or exclude channel 20						
			Excluded																									Exclude						
			Included																									Include						
V	RW		CH21																									Include or exclude channel 21						
			Excluded																									Exclude						
			Included																									Include						
W	RW		CH22																									Include or exclude channel 22						
			Excluded																									Exclude						
			Included																									Include						
X	RW		CH23																									Include or exclude channel 23						
			Excluded																									Exclude						
			Included																									Include						
Y	RW		CH24																									Include or exclude channel 24						
			Excluded																									Exclude						
			Included																									Include						
Z	RW																																	

22.2.49 CHG[5]

Address offset: 0x814

## Channel group 5

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
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		
<b>A RW CH0</b>																																		
			Excluded	0	Include or exclude channel 0																													
			Included	1	Exclude																													
<b>B RW CH1</b>																																		
			Excluded	0	Include or exclude channel 1																													
			Included	1	Exclude																													
<b>C RW CH2</b>																																		
			Excluded	0	Include or exclude channel 2																													
			Included	1	Exclude																													
<b>D RW CH3</b>																																		
			Excluded	0	Include or exclude channel 3																													
			Included	1	Exclude																													
<b>E RW CH4</b>																																		
			Excluded	0	Include or exclude channel 4																													
			Included	1	Exclude																													
<b>F RW CH5</b>																																		
			Excluded	0	Include or exclude channel 5																													
			Included	1	Exclude																													
<b>G RW CH6</b>																																		
			Excluded	0	Include or exclude channel 6																													
			Included	1	Exclude																													
<b>H RW CH7</b>																																		
			Excluded	0	Include or exclude channel 7																													
			Included	1	Exclude																													
<b>I RW CH8</b>																																		
			Excluded	0	Include or exclude channel 8																													
			Included	1	Exclude																													
<b>J RW CH9</b>																																		
			Excluded	0	Include or exclude channel 9																													
			Included	1	Exclude																													
<b>K RW CH10</b>																																		
			Excluded	0	Include or exclude channel 10																													
			Included	1	Exclude																													
<b>L RW CH11</b>																																		
			Excluded	0	Include or exclude channel 11																													
			Included	1	Exclude																													
<b>M RW CH12</b>																																		
			Excluded	0	Include or exclude channel 12																													
			Included	1	Exclude																													
<b>N RW CH13</b>																																		

Bit number			31	30	29	28	27	26	25	24	23	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		
<b>Id RW Field</b>			<b>Value Id</b>	<b>Value</b>																														
R RW CH17			Included	1																														
S RW CH18			Excluded	0																														
T RW CH19			Included	1																														
U RW CH20			Excluded	0																														
V RW CH21			Included	1																														
W RW CH22			Excluded	0																														
X RW CH23			Included	1																														
Y RW CH24			Excluded	0																														
Z RW CH25			Included	1																														
a RW CH26			Excluded	0																														
b RW CH27			Included	1																														
c RW CH28			Excluded	0																														
d RW CH29			Included	1																														
e RW CH30			Excluded	0																														
f RW CH31			Included	1																														

## 22.2.50 FORK[0].TEP

Address offset: 0x910

Channel 0 task end-point

## 22.2.51 FORK[1].TEP

Address offset: 0x914

## Channel 1 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	RW	TEP				Pointer to task register																										

## 22.2.52 FORK[2].TEP

Address offset: 0x918

## Channel 2 task end-point

## 22.2.53 FORK[3].TEP

Address offset: 0x91C

### Channel 3 task end-point

## 22.2.54 FORK[4].TEP

Address offset: 0x920

## Channel 4 task end-point

## 22.2.55 FORK[5].TEP

Address offset: 0x924

## Channel 5 task end-point

## 22.2.56 FORK[6].TEP

Address offset: 0x928

## Channel 6 task end-point

## 22.2.57 FORK[7].TEP

Address offset: 0x92C

## Channel 7 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	TEP				Pointer to task register																										

## 22.2.58 FORK[8].TEP

Address offset: 0x930

## Channel 8 task end-point

## 22.2.59 FORK[9].TEP

Address offset: 0x934

## Channel 9 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	RW	TEP				Pointer to task register																										

## 22.2.60 FORK[10].TEP

Address offset: 0x938

## Channel 10 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	0
Id	RW	Field	Value Id																								Description							
A	RW	TEP																									Pointer to task register							

## 22.2.61 FORK[11].TEP

Address offset: 0x93C

## Channel 11 task end-point

## 22.2.62 FORK[12].TEP

Address offset: 0x940

## Channel 12 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TEP		Pointer to task register																												

## 22.2.63 FORK[13].TEP

Address offset: 0x944

## Channel 13 task end-point

## 22.2.64 FORK[14].TEP

Address offset: 0x948

## Channel 14 task end-point

## 22.2.65 FORK[15].TEP

Address offset: 0x94C

## Channel 15 task end-point

## 22.2.66 FORK[16].TEP

Address offset: 0x950

## Channel 16 task end-point

## 22.2.67 FORK[17].TEP

Address offset: 0x954

Channel 17 task end-point

## 22.2.68 FORK[18].TEP

Address offset: 0x958

## Channel 18 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TEP			Pointer to task register																											

## 22.2.69 FORK[19].TEP

Address offset: 0x95C

Channel 19 task end-point

## 22.2.70 FORK[20].TEP

Address offset: 0x960

## Channel 20 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	RW	TEP				Pointer to task register																										

## 22.2.71 FORK[21].TEP

Address offset: 0x964

## Channel 21 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TEP			Pointer to task register																											

## 22.2.72 FORK[22].TEP

Address offset: 0x968

Channel 22 task end-point

## 22.2.73 FORK[23].TEP

Address offset: 0x96C

## Channel 23 task end-point

## 22.2.74 FORK[24].TEP

Address offset: 0x970

## Channel 24 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	
Id	RW	Field	Value Id																								Description							
A	RW	TEP																									Pointer to task register							

## 22.2.75 FORK[25].TEP

Address offset: 0x974

## Channel 25 task end-point

## 22.2.76 FORK[26].TEP

Address offset: 0x978

## Channel 26 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	0
Id	RW	Field	Value Id																								Description							
A	RW	TEP																									Pointer to task register							

## 22.2.77 FORK[27].TEP

Address offset: 0x97C

## Channel 27 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TEP			Pointer to task register																											

## 22.2.78 FORK[28].TEP

Address offset: 0x980

## Channel 28 task end-point

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TEP			Pointer to task register																											

## 22.2.79 FORK[29].TEP

Address offset: 0x984

## Channel 29 task end-point

## 22.2.80 FORK[30].TEP

Address offset: 0x988

## Channel 30 task end-point

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Id	RW	Field	Value Id																										Description					
A	RW	TEP																										Pointer to task register						

## 22.2.81 FORK[31].TEP

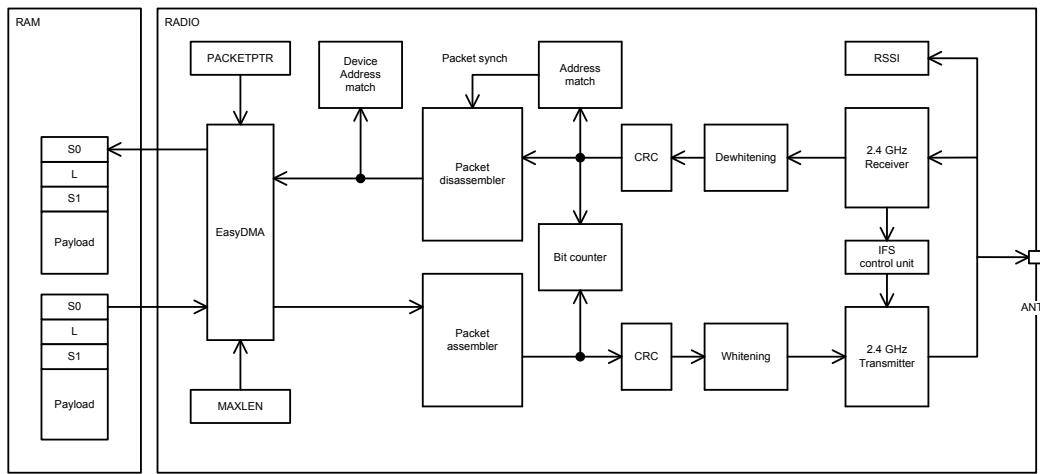
Address offset: 0x98C

## Channel 31 task end-point

## 23 RADIO — 2.4 GHz Radio

The RADIO contains a 2.4 GHz radio receiver and a 2.4 GHz radio transmitter that is compatible with Nordic's proprietary 1 Mbps and 2 Mbps radio modes in addition to 1 Mbps *Bluetooth®* low energy mode.

EasyDMA in combination with an automated packet assembler and packet disassembler, and an automated CRC generator and CRC checker, makes it very easy to configure and use the RADIO. See [Figure 29: RADIO block diagram](#) on page 204 for details.



**Figure 29: RADIO block diagram**

The RADIO includes a Device Address Match unit and an interframe spacing control unit that can be utilized to simplify address white listing and interframe spacing respectively, in *Bluetooth Smart* and similar applications.

The RADIO also includes a Received Signal Strength Indicator (RSSI) and a bit counter. The bit counter generates events when a preconfigured number of bits have been sent or received by the RADIO.

### 23.1 EasyDMA

The RADIO use EasyDMA for reading and writing of data packets from and to the RAM without CPU involvement.

As illustrated in [Figure 29: RADIO block diagram](#) on page 204, the RADIO's EasyDMA utilizes the same PACKETPTR for receiving and transmitting packets. The CPU should reconfigure this pointer every time before the RADIO is started via the START task.

The structure of a radio packet is described in detail in [Packet configuration](#) on page 205. The data that is stored in Data RAM and transported by EasyDMA consists of S0, LENGTH, S1, the payload itself, and a static add-on sent immediately after the payload.

The size of each of the above elements in the frame is configurable (see [Packet configuration](#) on page 205), and the space occupied in RAM depends on these settings. A size of zero is possible for any of the fields, it is up to the user to make sure that the resulting frame complies with the RF protocol chosen.

For the field sizes defined in bits, the occupation in RAM will always be rounded up to the next full byte size (for instance 3 bit length will allocate 1 byte in RAM, 9 bit length will allocate 2 bytes, etc.).

In addition, the S0INCL field in PCNF0 determines if S0 is present in RAM at all if its length is zero. If present, one byte is allocated in RAM.

The size of S0 is configured through the S0LEN field in PCNF0. The size of LENGTH is configured through the LFLEN field in PCNF0. The size of S1 is configured through the S1LEN field in PCNF0. The size of the payload is configured through the value in RAM corresponding to the LENGTH field. The size of the static add-on to the payload is configured through the STATLEN field in PCNF1.

The MAXLEN field in the PCNF1 register configures the maximum packet payload plus add-on size in number of bytes that can be transmitted or received by the RADIO. This feature can be used to ensure that the RADIO does not overwrite, or read beyond, the RAM assigned to the packet payload. This means that if the packet payload length defined by PCNF1.STATLEN and the LENGTH field in the packet specifies a packet larger than MAXLEN, the payload will be truncated at MAXLEN.

Note that MAXLEN includes the payload and the add-on, but excludes the size occupied by the S0, LENGTH and S1 fields. This has to be taken into account when allocating RAM.

If the payload plus add-on length is specified larger than MAXLEN, the RADIO will still transmit or receive in the same way as before except the payload is now truncated to MAXLEN. The packet's LENGTH field will not be altered when the payload is truncated. The RADIO will calculate CRC as if the packet length is equal to MAXLEN.

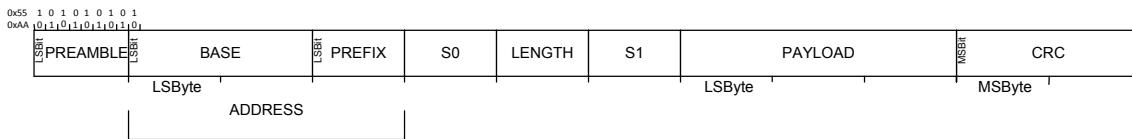
If the PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

The DISABLED event indicates that the EasyDMA has finished accessing the RAM.

## 23.2 Packet configuration

A Radio packet contains the following fields: PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD and CRC.

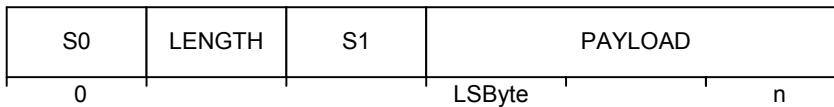
See [Figure 30: On-air packet layout](#) on page 205. Not shown in the figure is the static payload add-on (the length of which is defined in STATLEN, and which is 0 bytes long in a standard BLE packet), and would be sent between PAYLOAD and CRC. The Radio sends the different fields in the packet in the order they are illustrated below, from left to right. The preamble will be sent least significant bit first on-air.



**Figure 30: On-air packet layout**

For all modes that can be specified in the MODE register, the PREAMBLE is one byte long. If the first bit of the ADDRESS is 0 the preamble will be set to 0xAA otherwise the PREAMBLE will be set to 0x55.

Radio packets are stored in memory inside instances of a radio packet data structure as illustrated in [Figure 31: In-RAM representation of radio packet, S0, LENGTH and S1 are optional](#) on page 205. The PREAMBLE, ADDRESS and CRC fields are omitted in this data structure.



**Figure 31: In-RAM representation of radio packet, S0, LENGTH and S1 are optional**

The byte ordering on air is always Least Significant Byte First for the ADDRESS and PAYLOAD fields and Most Significant Byte First for the CRC field. The ADDRESS fields are always transmitted and received least significant bit first on-air. The CRC field is always transmitted and received Most Significant Bit first. The bit-

endian, i.e. which order the bits are sent and received in, of the S0, LENGTH, S1 and PAYLOAD fields can be configured via the ENDIAN in PCNF1.

The S0INCL field in PCNF0 determines if S0 is present in RAM at all if its length is zero. If present, one byte is allocated in RAM.

The sizes of the S0, LENGTH and S1 fields can be individually configured via S0LEN, Lflen and S1LEN in PCNF0 respectively. If any of these fields are configured to be less than 8 bit long the, the least significant bits of the fields, as seen from the RAM representation, are used.

If S0, LENGTH or S1 are specified with zero length their fields will be omitted in memory, otherwise each field will be represented as a separate byte, regardless of the number of bits in their on-air counterpart.

## 23.3 Maximum packet length

Independent of the configuration of MAXLEN, the combined length of S0, LENGTH, S1 and PAYLOAD cannot exceed 258 bytes.

## 23.4 Address configuration

The on-air radio ADDRESS field is composed of two parts, the base address field and the address prefix field.

The size of the base address field is configurable via BALEN in PCNF1. The base address is truncated from LSByte if the BALEN is less than 4. See [Table 38: Definition of logical addresses](#) on page 206.

The on-air addresses are defined in the BASEn and PREFIXn registers, and it is only when writing these registers the user will have to relate to actual on-air addresses. For other radio address registers such as the TXADDRESS, RXADDRESSES and RXMATCH registers, logical radio addresses ranging from 0 to 7 are being used. The relationship between the on-air radio addresses and the logical addresses is described in [Table 38: Definition of logical addresses](#) on page 206.

**Table 38: Definition of logical addresses**

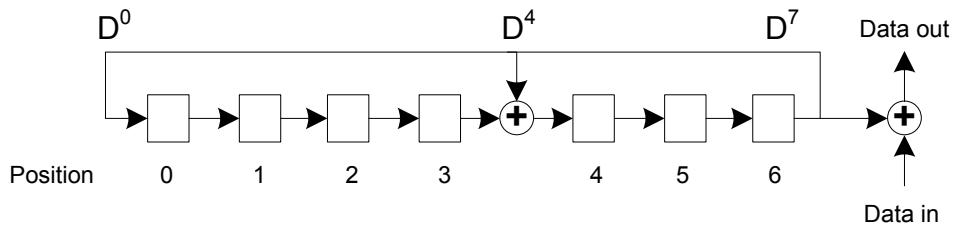
Logical address	Base address	Prefix byte
0	BASE0	PREFIX0.AP0
1	BASE1	PREFIX0.AP1
2	BASE1	PREFIX0.AP2
3	BASE1	PREFIX0.AP3
4	BASE1	PREFIX1.AP4
5	BASE1	PREFIX1.AP5
6	BASE1	PREFIX1.AP6
7	BASE1	PREFIX1.AP7

## 23.5 Data whitening

The RADIO is able to do packet whitening and de-whitening.

See WHITEEN in PCNF1 register for how to enable whitening. When enabled, whitening and de-whitening will be handled by the RADIO automatically as packets are sent and received.

The whitening word is generated using polynomial  $g(D) = D^7 + D^4 + 1$ , which then is XORed with the data packet that is to be whitened, or de-whitened. See the figure below.



**Figure 32: Data whitening and de-whitening**

Whitening and de-whitening will be performed over the whole packet (except for the preamble and the address field).

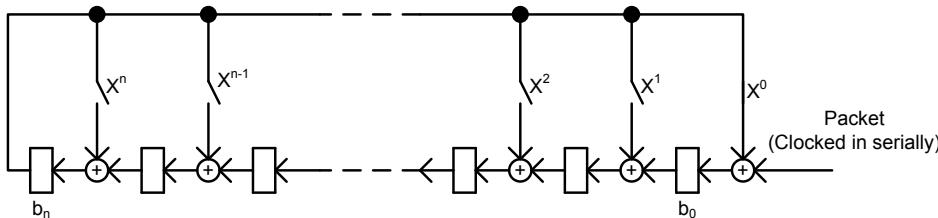
The linear feedback shift register, illustrated in [Figure 32: Data whitening and de-whitening](#) on page 207 can be initialised via the DATAWHITEIV register.

## 23.6 CRC

The CRC generator in the RADIO calculates the CRC over the whole packet excluding the preamble. If desirable, the address field can be excluded from the CRC calculation as well

See CRCCNF register for more information.

The CRC polynomial is configurable as illustrated in [Figure 33: CRC generation of an n bit CRC](#) on page 207 where bit 0 in the CRCPOLY register corresponds to  $X^0$  and bit 1 corresponds to  $X^1$  etc. See CRCPOLY for more information.



**Figure 33: CRC generation of an n bit CRC**

As illustrated in [Figure 33: CRC generation of an n bit CRC](#) on page 207, the CRC is calculated by feeding the packet serially through the CRC generator. Before the packet is clocked through the CRC generator, the CRC generator's latches  $b_0$  through  $b_n$  will be initialized with a predefined value specified in the CRCINIT register. When the whole packet is clocked through the CRC generator, latches  $b_0$  through  $b_n$  will hold the resulting CRC. This value will be used by the RADIO during both transmission and reception but it is not available to be read by the CPU at any time. A received CRC can however be read by the CPU via the RXCRC register independent of whether or not it has passed the CRC check.

The length ( $n$ ) of the CRC is configurable, see CRCCNF for more information.

After the whole packet including the CRC has been received, the RADIO will generate a CRCOK event if no CRC errors were detected, or alternatively generate a CRCERROR event if CRC errors were detected.

The status of the CRC check can be read from the **CRCSTATUS** register after a packet has been received.

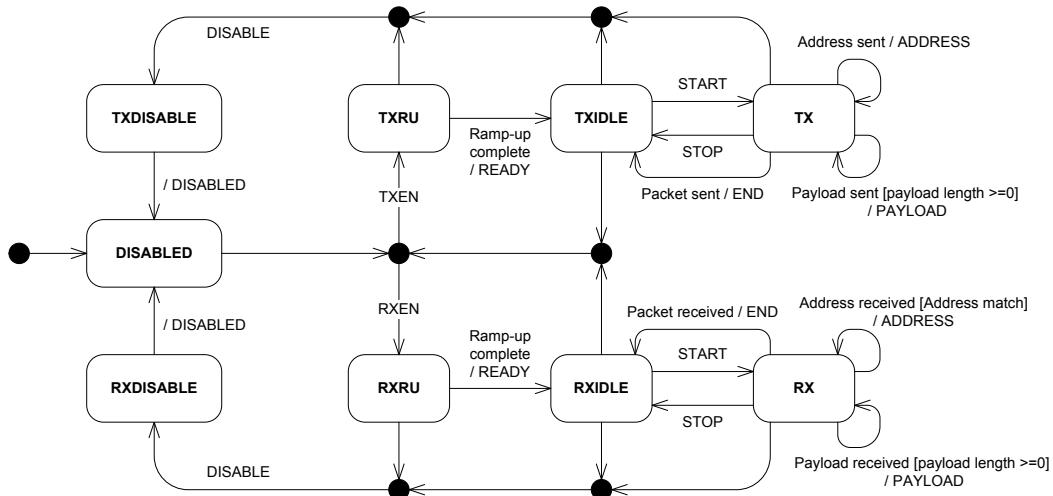
## 23.7 Radio states

The RADIO can enter a number of states.

The RADIO can enter the states described the table below. An overview state diagram for the RADIO is illustrated in [Figure 34: Radio states](#) on page 208. This figure shows how the tasks and events relate to the RADIO's operation. The RADIO does not prevent a task from being triggered from the wrong state. If a task is triggered from the wrong state, for example if the RXEN task is triggered from the RXDISABLE state, this may lead to incorrect behaviour. As illustrated in [Figure 34: Radio states](#) on page 208, the PAYLOAD event is always generated even if the payload is zero.

**Table 39: RADIO state diagram**

State	Description
DISABLED	No operations are going on inside the radio and the power consumption is at a minimum
RXRU	The radio is ramping up and preparing for reception
RXIDLE	The radio is ready for reception to start
RX	Reception has been started and the addresses enabled in the RXADDRESSES register are being monitored
TXRU	The radio is ramping up and preparing for transmission
TXIDLE	The radio is ready for transmission to start
TX	The radio is transmitting a packet
RXDISABLE	The radio is disabling the receiver
TXDISABLE	The radio is disabling the transmitter



**Figure 34: Radio states**

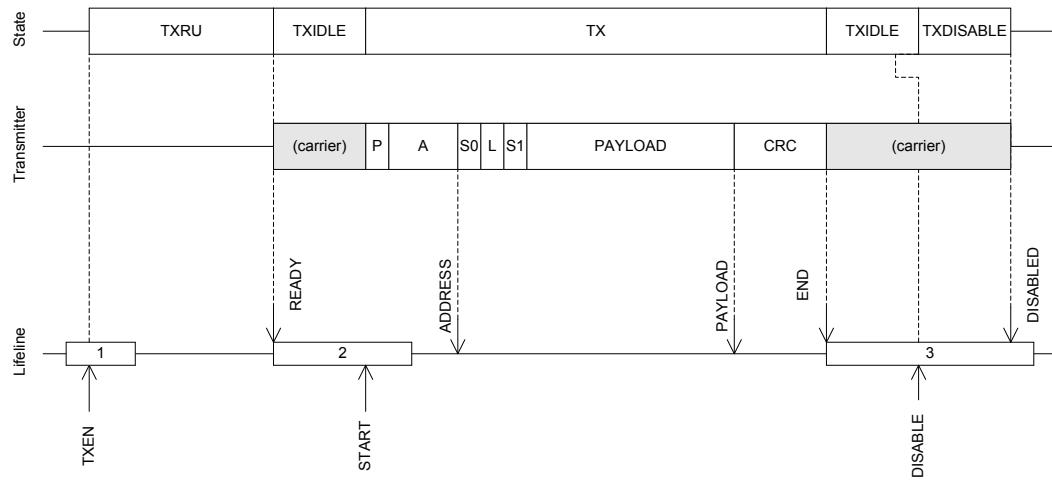
## 23.8 Transmit sequence

Before the RADIO is able to transmit a packet, it must first ramp-up in TX mode.

See TXRU in [Figure 34: Radio states](#) on page 208 and [Figure 35: Transmit sequence](#) on page 209. A TXRU ramp-up sequence is initiated when the TXEN task is triggered. After the radio has successfully ramped up it will generate the READY event indicating that a packet transmission can be initiate. A packet transmission is initiated by triggering the START task. As illustrated in [Figure 34: Radio states](#) on page 208 the START task can first be triggered after the RADIO has entered into the TXIDLE state.

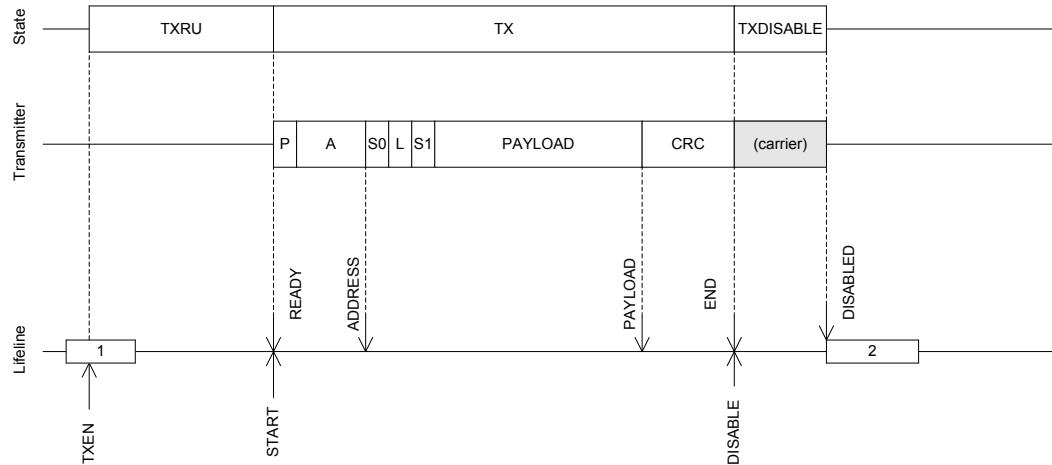
[Figure 35: Transmit sequence](#) on page 209 illustrates a single packet transmission where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay caused by CPU execution is expected between READY and START, and between END and DISABLE. As illustrated in [Figure 35: Transmit sequence](#) on page 209

the RADIO will by default transmit '1's between READY and START, and between END and DISABLED. What is transmitted can be programmed through the DTX field in the MODECNF0 register.



**Figure 35: Transmit sequence**

A slightly modified version of the transmit sequence from [Figure 35: Transmit sequence](#) on page 209 is illustrated in [Figure 36: Transmit sequence using shortcuts to avoid delays](#) on page 209 where the RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.



**Figure 36: Transmit sequence using shortcuts to avoid delays**

The RADIO is able to send multiple packets one after the other without having to disable and re-enable the RADIO between packets, this is illustrated in [Figure 37: Transmission of multiple packets](#) on page 210.

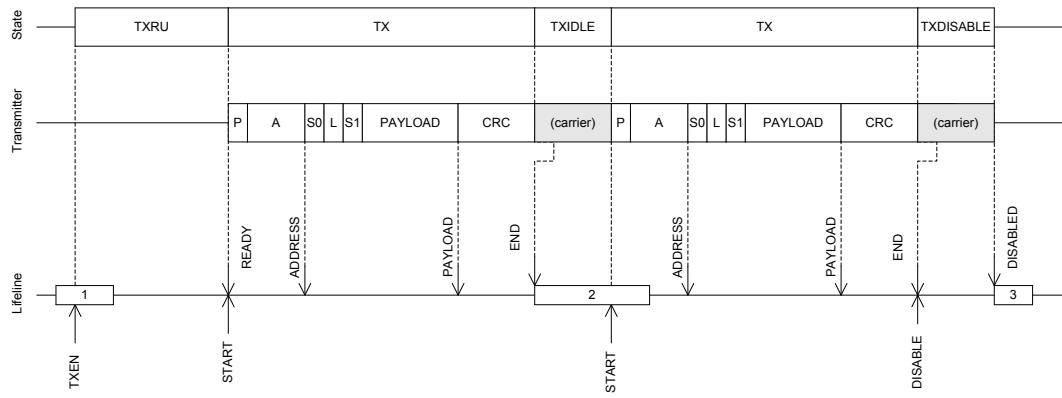


Figure 37: Transmission of multiple packets

## 23.9 Receive sequence

Before the RADIO is able to receive a packet, it must first ramp up in RX mode

See RXRU in [Figure 34: Radio states](#) on page 208 and [Figure 38: Receive sequence](#) on page 210. An RXRU ramp-up sequence is initiated when the RXEN task is triggered. After the radio has successfully ramped up it will generate the READY event indicating that a packet reception can be initiated. A packet reception is initiated by triggering the START task. As illustrated in [Figure 34: Radio states](#) on page 208 the START task can, first be triggered after the RADIO has entered into the RXIDLE state.

[Figure 38: Receive sequence](#) on page 210 illustrates a single packet reception where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay, caused by CPU execution, is expected between READY and START, and between END and DISABLE. As illustrated [Figure 38: Receive sequence](#) on page 210 the RADIO will be listening and possibly receiving undefined data, illustrated with an 'X', from START and until a packet with valid preamble (P) is received.

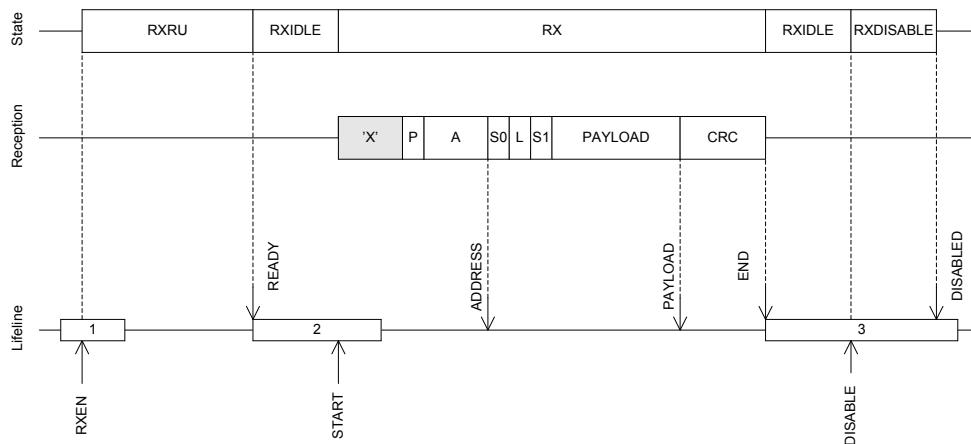
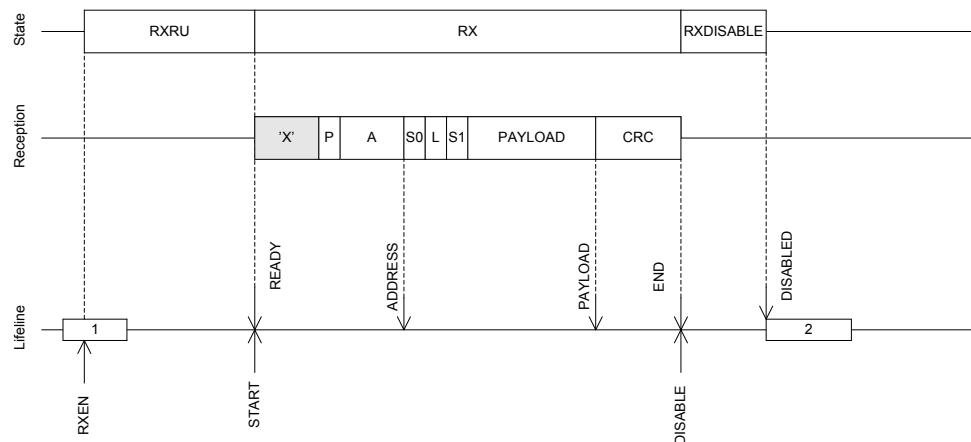


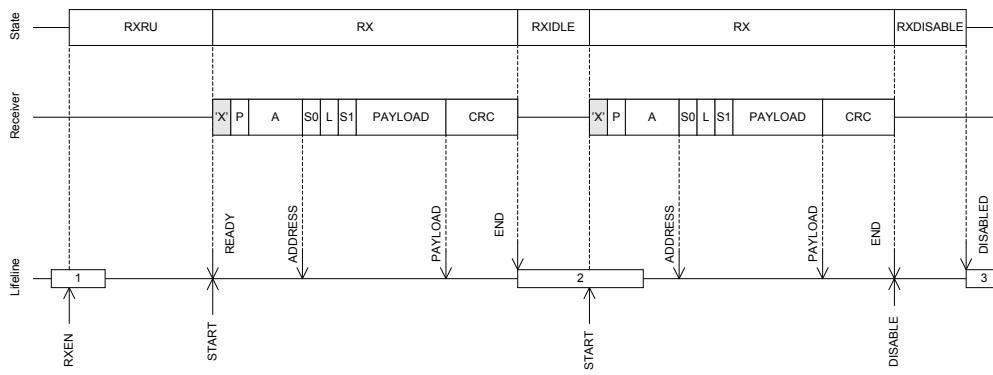
Figure 38: Receive sequence

A slightly modified version of the receive sequence from [Figure 38: Receive sequence](#) on page 210 is illustrated in [Figure 39: Receive sequence using shortcuts to avoid delays](#) on page 211 where the the RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.



**Figure 39: Receive sequence using shortcuts to avoid delays**

The RADIO is able to receive multiple packets one after the other without having to disable and re-enable the RADIO between packets, this is illustrated [Figure 40: Reception of multiple packets](#) on page 211.



**Figure 40: Reception of multiple packets**

## 23.10 Received Signal Strength Indicator (RSSI)

The radio implements a mechanism for measuring the power in the received radio signal. This feature is called Received Signal Strength Indicator (RSSI).

Sampling of the received signal strength is started by using the RSSISTART task. The sample can be read from the RSSISAMPLE register.

The sample period of the RSSI is defined by  $\text{RSSI}_{\text{PERIOD}}$ , see the device product specification for details. The RSSI sample will hold the average received signal strength during this sample period.

For the RSSI sample to be valid the radio has to be enabled in receive mode (RXEN task) and the reception has to be started (READY event followed by START task).

## 23.11 Interframe spacing

Interframe spacing is the time interval between two consecutive packets.

It is defined as the time, in micro seconds, from the end of the last bit of the previous packet received and to the start of the first bit of the subsequent packet that is transmitted. The RADIO is able to enforce this

interval as specified in the TIFS register as long as TIFS is not specified to be shorter than the RADIO's turn-around time, i.e. the time needed to switch off the receiver, and switch back on the transmitter.

TIFS is only enforced if END\_DISABLE and DISABLED\_TXEN or END\_DISABLE and DISABLED\_RXEN shortcuts are enabled. TIFS is only qualified for use in BLE\_1MBIT mode, and default ramp-up mode.

## 23.12 Device address match

The device address match feature is tailored for address white listing in a Bluetooth Smart and similar implementations.

This feature enables on-the-fly device address matching while receiving a packet on air. This feature only works in receive mode and as long as RADIO is configured for little endian, see PCNF1.ENDIAN.

The Device Address match unit assumes that the 48 first bits of the payload is the device address and that bit number 6 in S0 is the TxAdd bit. See the Bluetooth Core Specification for more information about device addresses, TxAdd and whitelisting.

The RADIO is able to listen for eight different device addresses at the same time. These addresses are specified in a DAB/DAP register pair, one pair per address, in addition to a TxAdd bit configured in the DACNF register. The DAB register specifies the 32 least significant bits of the device address, while the DAP register specifies the 16 most significant bits of the device address.

Each of the device addresses can be individually included or excluded from the matching mechanism. This is configured in the DACNF register.

## 23.13 Bit counter

The RADIO implements a simple counter that can be configured to generate an event after a specific number of bits have been transmitted or received.

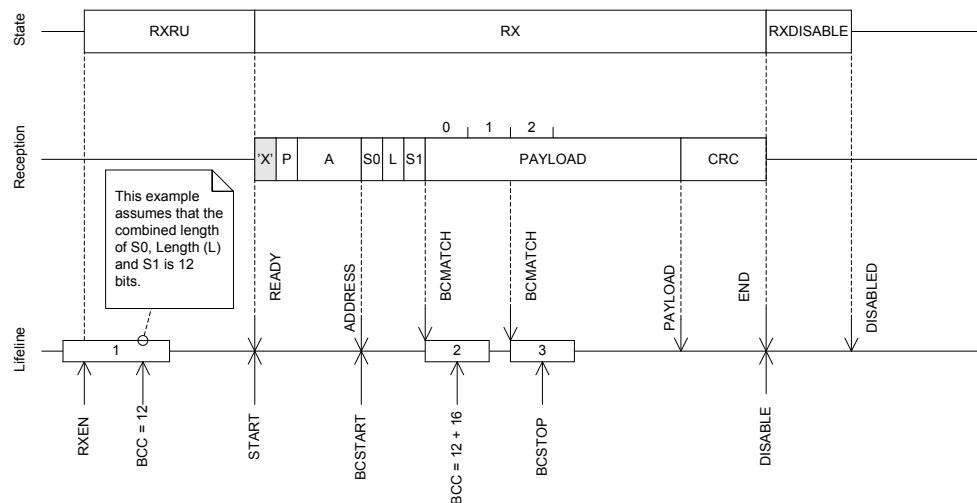
By using shortcuts, this counter can be started from different events generated by the RADIO and hence count relative to these.

The bit counter is started by triggering the BCSTART task, and stopped by triggering the BCSTOP task. A BCMATCH event will be generated when the bit counter has counted the number of bits specified in the BCC register. The bit counter will continue to count bits until the DISABLED event is generated or until the BCSTOP task is triggered. The CPU can therefore, after a BCMATCH event, reconfigure the BCC value for new BCMATCH events within the same packet.

The bit counter can only be started after the RADIO has received the ADDRESS event.

The bit counter will stop and reset on BCSTOP, STOP, END and DISABLE tasks.

The figure below illustrates how the bit counter can be used to generate a BCMATCH event in the beginning of the packet payload, and again generate a second BCMATCH event after sending 2 bytes (16 bits) of the payload.



**Figure 41: Bit counter example**

## 23.14 Registers

**Table 40: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40001000	RADIO	RADIO	2.4 GHz radio	

**Table 41: Register Overview**

Register	Offset	Description
TASKS_TXEN	0x000	Enable RADIO in TX mode
TASKS_RXEN	0x004	Enable RADIO in RX mode
TASKS_START	0x008	Start RADIO
TASKS_STOP	0x00C	Stop RADIO
TASKS_DISABLE	0x010	Disable RADIO
TASKS_RSSISTART	0x014	Start the RSSI and take one single sample of the receive signal strength.
TASKS_RSSISTOP	0x018	Stop the RSSI measurement
TASKS_BCSTART	0x01C	Start the bit counter
TASKS_BCSTOP	0x020	Stop the bit counter
EVENTS_READY	0x100	RADIO has ramped up and is ready to be started
EVENTS_ADDRESS	0x104	Address sent or received
EVENTS_PAYLOAD	0x108	Packet payload sent or received
EVENTS_END	0x10C	Packet sent or received
EVENTS_DISABLED	0x110	RADIO has been disabled
EVENTS_DEVMATCH	0x114	A device address match occurred on the last received packet
EVENTS_DEVMISS	0x118	No device address match occurred on the last received packet
EVENTS_RSSIEND	0x11C	Sampling of receive signal strength complete.
EVENTS_BCMATCH	0x128	Bit counter reached bit count value.
EVENTS_CRCOK	0x130	Packet received with CRC ok
EVENTS_CRCERROR	0x134	Packet received with CRC error
<i>SHORTS</i>	0x200	Shortcut register
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>CRCSTATUS</i>	0x400	CRC status
<i>RXMATCH</i>	0x408	Received address
<i>RXCRC</i>	0x40C	CRC field of previously received packet

Register	Offset	Description
<i>DAI</i>	0x410	Device address match index
<i>PACKETPTR</i>	0x504	Packet pointer
<i>FREQUENCY</i>	0x508	Frequency
<i>TXPOWER</i>	0x50C	Output power
<i>MODE</i>	0x510	Data rate and modulation
<i>PCNFO</i>	0x514	Packet configuration register 0
<i>PCNF1</i>	0x518	Packet configuration register 1
<i>BASE0</i>	0x51C	Base address 0
<i>BASE1</i>	0x520	Base address 1
<i>PREFIX0</i>	0x524	Prefixes bytes for logical addresses 0-3
<i>PREFIX1</i>	0x528	Prefixes bytes for logical addresses 4-7
<i>TXADDRESS</i>	0x52C	Transmit address select
<i>RXADDRESSES</i>	0x530	Receive address select
<i>CRCCNF</i>	0x534	CRC configuration
<i>CRCPOLY</i>	0x538	CRC polynomial
<i>CRCINIT</i>	0x53C	CRC initial value
	0x540	Reserved
<i>TIFS</i>	0x544	Inter Frame Spacing in us
<i>RSSISAMPLE</i>	0x548	RSSI sample
<i>STATE</i>	0x550	Current radio state
<i>DATAWHITEIV</i>	0x554	Data whitening initial value
<i>BCC</i>	0x560	Bit counter compare
<i>DAB[0]</i>	0x600	Device address base segment 0
<i>DAB[1]</i>	0x604	Device address base segment 1
<i>DAB[2]</i>	0x608	Device address base segment 2
<i>DAB[3]</i>	0x60C	Device address base segment 3
<i>DAB[4]</i>	0x610	Device address base segment 4
<i>DAB[5]</i>	0x614	Device address base segment 5
<i>DAB[6]</i>	0x618	Device address base segment 6
<i>DAB[7]</i>	0x61C	Device address base segment 7
<i>DAP[0]</i>	0x620	Device address prefix 0
<i>DAP[1]</i>	0x624	Device address prefix 1
<i>DAP[2]</i>	0x628	Device address prefix 2
<i>DAP[3]</i>	0x62C	Device address prefix 3
<i>DAP[4]</i>	0x630	Device address prefix 4
<i>DAP[5]</i>	0x634	Device address prefix 5
<i>DAP[6]</i>	0x638	Device address prefix 6
<i>DAP[7]</i>	0x63C	Device address prefix 7
<i>DACNF</i>	0x640	Device address match configuration
<i>MODECINFO</i>	0x650	Radio mode configuration register 0
<i>POWER</i>	0xFFC	Peripheral power control

## 23.14.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																										H	G	F	E	D	C	B	A
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																											
A	RW	READY_START				Shortcut between READY event and START task																											
						See <a href="#">EVENTS_READY</a> and <a href="#">TASKS_START</a>																											
		Disabled	0			Disable shortcut																											
		Enabled	1			Enable shortcut																											
B	RW	END_DISABLE				Shortcut between END event and DISABLE task																											
						See <a href="#">EVENTS_END</a> and <a href="#">TASKS_DISABLE</a>																											

## 23.14.2 INTENSET

Address offset: 0x304

## Enable interrupt

### 23.14.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																				L	K	I		H	G	F	E	D	C	B	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	READY			Write '1' to Disable interrupt for READY event																											
					See <a href="#">EVENTS_READY</a>																											
		Clear	1	Disable																												

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	L K I H G F E D C B A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Description</b>
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
B RW ADDRESS					Write '1' to Disable interrupt for ADDRESS event
					See <a href="#">EVENTS_ADDRESS</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
C RW PAYLOAD					Write '1' to Disable interrupt for PAYLOAD event
					See <a href="#">EVENTS_PAYLOAD</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
D RW END					Write '1' to Disable interrupt for END event
					See <a href="#">EVENTS_END</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
E RW DISABLED					Write '1' to Disable interrupt for DISABLED event
					See <a href="#">EVENTS_DISABLED</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
F RW DEVMATCH					Write '1' to Disable interrupt for DEVMATCH event
					See <a href="#">EVENTS_DEVMATCH</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
G RW DEVMISS					Write '1' to Disable interrupt for DEVMISS event
					See <a href="#">EVENTS_DEVMISS</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
H RW RSSIEND					Write '1' to Disable interrupt for RSSIEND event
					See <a href="#">EVENTS_RSSIEND</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
I RW BCMATCH					Write '1' to Disable interrupt for BCMATCH event
					See <a href="#">EVENTS_BCMATCH</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
K RW CRCOK					Write '1' to Disable interrupt for CRCOK event
					See <a href="#">EVENTS_CRCOK</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled
		Enabled	1		Read: Enabled
L RW CRCERROR					Write '1' to Disable interrupt for CRCERROR event
					See <a href="#">EVENTS_CRCERROR</a>
		Clear	1		Disable
		Disabled	0		Read: Disabled

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																			L	K	I		H	G	F	E	D	C	B	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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																												

## 23.14.4 CRCSTATUS

Address offset: 0x400

## CRC status

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	R	CRCSTATUS				CRC status of packet received																										
			CRCError	0		Packet received with CRC error																										
			CRCOk	1		Packet received with CRC ok																										

## 23.14.5 RXMATCH

Address offset: 0x408

## Received address

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value	Description																											
A	R	RXMATCH			Received address																											
					Logical address of which previous packet was received																											

## 23.14.6 RXCRC

Address offset: 0x40C

CRC field of previously received packet

23.14.7 DAI

Address offset: 0x410

## Device address match index

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A																													
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	R	DAI				Device address match index																										
						Index (n) of device address, see DAB[n] and DAP[n], that got an address match.																										

## 23.14.8 PACKETPTR

Address offset: 0x504

## Packet pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	0	
Id	RW	Field	Value																								Description								
A	RW	PACKETPTR																									Packet pointer								
																											Packet address to be used for the next transmission or reception. When transmitting, the packet pointed to by this address will be transmitted and when receiving, the received packet will be written to this address. This address is a byte aligned ram address.								

## **23.14.9 FREQUENCY**

Address offset: 0x508

## Frequency

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000002</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Id	RW	Field	Value Id	Value	Description																											
A	RW	FREQUENCY		[0..100]	Radio channel frequency  Frequency = 2400 + FREQUENCY (MHz).																											
B	RW	MAP			Channel map selection.																											
		Default	0		Channel map between 2400 MHZ .. 2500 MHz																											
					Frequency = 2400 + FREQUENCY (MHz)																											
		Low	1		Channel map between 2360 MHZ .. 2460 MHz																											
					Frequency = 2360 + FREQUENCY (MHz)																											

## 23.14.10 TXPOWER

Address offset: 0x50C

## Output power

## **23.14.11 MODE**

Address offset: 0x510

## Data rate and modulation

23.14.12 PCNF0

Address offset: 0x514

## Packet configuration register 0

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id															G	F	E	E	E	C		A	A	A	A							
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id																								Description					
A	RW	LFLEN																									Length on air of LENGTH field in number of bits.					
C	RW	SOLEN																									Length on air of S0 field in number of bytes.					
E	RW	S1LEN																									Length on air of S1 field in number of bits.					
F	RW	S1INCL																									Include or exclude S1 field in RAM					
G	RW	PLEN																									Length of preamble on air. Decision point: TASKS_START task					
																											8bit					
H	RW	PSEL																									8-bit preamble					
																											16bit					
I	RW	PSEL																									16-bit preamble					

23.14.13 PCNF1

Address offset: 0x518

## Packet configuration register 1

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id								E	D			C	C	C	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																										
A	RW	MAXLEN	[0..255]			Maximum length of packet payload. If the packet payload is larger than MAXLEN, the radio will truncate the payload to MAXLEN.																										
B	RW	STATLEN	[0..255]			Static length in number of bytes  The static length parameter is added to the total length of the payload when sending and receiving packets, e.g. if the static length is set to N the radio will receive or send N bytes more than what is defined in the LENGTH field of the packet.																										
C	RW	BALEN	[2..4]			Base address length in number of bytes  The address field is composed of the base address and the one byte long address prefix, e.g. set BALEN=2 to get a total address of 3 bytes.																										
D	RW	ENDIAN				On air endianness of packet, this applies to the S0, LENGTH, S1 and the PAYLOAD fields.  Little Big																										
			0			Least Significant bit on air first																										
			1			Most significant bit on air first																										
E	RW	WHITEEN				Enable or disable packet whitening  Disabled Enabled																										
			0			Disable																										
			1			Enable																										

## 23.14.14 BASE0

Address offset: 0x51C

Base address 0

23.14.15 BASE1

Address offset: 0x520

Base address 1

## 23.14.16 PREFIXO

Address offset: 0x524

Prefixes bytes for logical addresses 0-3

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	D	D	D	D	D	D	C	C	C	C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>		
Id	RW	Field	Value Id	Value		Description																										
A	RW	AP0				Address prefix 0.																										
B	RW	AP1				Address prefix 1.																										
C	RW	AP2				Address prefix 2.																										
D	RW	AP3				Address prefix 3.																										

## 23.14.17 PREFIX1

Address offset: 0x528

## Prefixes bytes for logical addresses 4-7

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	D	D	D	D	D	D	C	C	C	C	C	C	C	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>		
Id	RW	Field	Value Id	Value		Description																										
A	RW	AP4				Address prefix 4.																										
B	RW	AP5				Address prefix 5.																										
C	RW	AP6				Address prefix 6.																										
D	RW	AP7				Address prefix 7.																										

## **23.14.18 TXADDRESS**

Address offset: 0x52C

## Transmit address select

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																														A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	TXADDRESS				Transmit address select																										
						Logical address to be used when transmitting a packet.																										

## 23.14.19 RXADDRESSES

Address offset: 0x530

## Receive address select

23.14.20 CRCCNF

Address offset: 0x534

## CRC configuration

## 23.14.21 CRCPOLY

Address offset: 0x538

## CRC polynomial

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	CRCPOLY				CRC polynomial																										

Each term in the CRC polynomial is mapped to a bit in this register which index corresponds to the term's exponent. The least significant term/bit is hard-wired internally to 1, and bit number 0 of the register content is ignored by the hardware.

The following example is for an 8 bit CRC polynomial:  $x^8 + x^7 + x^3 + x^2 + 1 = 1\ 1000\ 1101$ .

23.14.22 CRCINIT

Address offset: 0x53C

## CRC initial value

Initial value for CRC calculation.

23.14.23 TIFS

Address offset: 0x544

### Inter Frame Spacing in us

Inter frame space is the time interval between two consecutive packets. It is defined as the time, in micro seconds, from the end of the last bit of the previous packet to the start of the first bit of the subsequent packet.

## 23.14.24 RSSISAMPLE

Address offset: 0x548

## RSSI sample

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																											A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	RSSISAMPLE	[0..127]	RSSI sample																												

RSSI sample result. The value of this register is read as a positive value while the actual received signal strength is a negative value. Actual received signal strength is therefore as follows:

$$\text{received signal strength} = -A \text{ dBm}$$

23.14.25 STATE

Address offset: 0x550

## Current radio state

23.14.26 DATAWHITEIV

Address offset: 0x554

## Data whitening initial value

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																																	
<b>Reset 0x00000040</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																												
A	RW	DATAWHITEIV			Data whitening initial value. Bit 6 is hard-wired to '1', writing '0' to it has no effect, and it will always be read back and used by the device as '1'.  Bit 0 corresponds to Position 6 of the LSFR, Bit 1 to Position 5, etc.																												

23.14.27 BCC

Address offset: 0x560

## Bit counter compare

23.14.28 DAB[0]

Address offset: 0x600

## Device address base segment 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		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	DAB			Device address base segment 0																											

### 23.14.29 DAB[1]

Address offset: 0x604

### Device address base segment 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	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	0	
Id	RW	Field	Value																								Description							
A	RW	DAB																									Device address base segment 1							

### 23.14.30 DAB[2]

Address offset: 0x608

## Device address base segment 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	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																									Description						
A	RW	DAB																									Device address base segment 2							

### 23.14.31 DAB[3]

Address offset: 0x60C

### Device address base segment 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	DAB			Device address base segment 3																											

### 23.14.32 DAB[4]

Address offset: 0x610

#### Device address base segment 4

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	DAB			Device address base segment 4																											

## 23.14.33 DAB[5]

Address offset: 0x614

Device address base segment 5

23.14.34 DAB[6]

Address offset: 0x618

## Device address base segment 6

23.14.35 DAB[7]

Address offset: 0x61C

## Device address base segment 7

23.14.36 DAP[0]

Address offset: 0x620

## Device address prefix 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	DAP			Device address prefix 0																											

23.14.37 DAP[1]

Address offset: 0x624

## Device address prefix 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		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	DAP			Device address prefix 1																											

23.14.38 DAP[2]

Address offset: 0x628

## Device address prefix 2

23.14.39 DAP[3]

Address offset: 0x62C

### Device address prefix 3

23.14.40 DAP[4]

Address offset: 0x630

## Device address prefix 4

### 23.14.41 DAP[5]

Address offset: 0x634

## Device address prefix 5

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	DAP			Device address prefix 5																											

23.14.42 DAP[6]

Address offset: 0x638

## Device address prefix 6

23.14.43 DAP[7]

Address offset: 0x63C

## Device address prefix 7

23.14.44 DACNF

Address offset: 0x640

## Device address match configuration

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																		P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id																								Description						
A	RW	ENAO																									Enable or disable device address matching using device address						
																											0						
																											Disabled						
B	RW	ENAO1																									Disabled						
																											1						
																											Enabled						
B	RW	ENAO1																									Enable or disable device address matching using device address						
																											1						
																											Disabled						

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id	P O N M L K J I H G F E D C B A		
<b>Reset 0x00000000</b>	<b>0 0</b>		
<b>Id RW Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
C RW ENA2	Enabled	1	Enabled
			Enable or disable device address matching using device address 2
D RW ENA3	Disabled	0	Disabled
	Enabled	1	Enabled
E RW ENA4	Disabled	0	Disabled
	Enabled	1	Enabled
F RW ENA5	Disabled	0	Disabled
	Enabled	1	Enabled
G RW ENA6	Disabled	0	Disabled
	Enabled	1	Enabled
H RW ENA7	Disabled	0	Disabled
	Enabled	1	Enabled
I RW TXADD0			TxAdd for device address 0
J RW TXADD1			TxAdd for device address 1
K RW TXADD2			TxAdd for device address 2
L RW TXADD3			TxAdd for device address 3
M RW TXADD4			TxAdd for device address 4
N RW TXADD5			TxAdd for device address 5
O RW TXADD6			TxAdd for device address 6
P RW TXADD7			TxAdd for device address 7

## 23.14.45 MODECNFO

Address offset: 0x650

Radio mode configuration register 0

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id	C C A		
<b>Reset 0x00000200</b>	<b>0 0</b>		
<b>Id RW Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A RW RU	Default	0	Radio ramp-up time Default ramp-up time (tRXEN), compatible with firmware written for nRF51
	Fast	1	Fast ramp-up (tRXEN,FAST), see electrical specification for more information
C RW DTX			Default TX value Specifies what the RADIO will transmit when it is not started, i.e. between: RADIO.EVENTS_READY and RADIO.TASKS_START RADIO.EVENTS_END and RADIO.TASKS_START

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Id	C C A
<b>Reset 0x000000200</b>	<b>0 1 0</b>
<b>Id RW Field Value Id Value Description</b>	
B1 0 RADIO.EVENTS_END and RADIO.EVENTS_DISABLED	
B0 1 Transmit '1'	
Center 2 Transmit '0'	
Transmit center frequency	
When tuning the crystal for centre frequency, the RADIO must be set in DTX = Center mode to be able to achieve the expected accuracy.	

## 23.14.46 POWER

Address offset: 0xFFC

Peripheral power control

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Id	A
<b>Reset 0x00000001</b>	<b>0 1</b>
<b>Id RW Field Value Id Value Description</b>	
A RW POWER Peripheral power control. The peripheral and its registers will be reset to its initial state by switching the peripheral off and then back on again.	
Disabled 0 Peripheral is powered off	
Enabled 1 Peripheral is powered on	

## 23.15 Electrical specification

### 23.15.1 General Radio Characteristics

Symbol	Description	Min.	Typ.	Max.	Units
$f_{OP}$	Operating frequencies	2360	2500	MHz	
$f_{PLL,PROG,RES}$	PLL programming resolution	2		kHz	
$f_{PLL,CH,SP}$	PLL channel spacing	1		MHz	
$f_{\Delta,1M}$	Frequency deviation @ 1 Msps	$\pm 170$		kHz	
$f_{\Delta,BLE,1M}$	Frequency deviation @ BLE 1Msps	$\pm 250$		kHz	
$f_{\Delta,2M}$	Frequency deviation @ 2 Msps	$\pm 320$		kHz	
$fsk_{SPS}$	On-the-air data rate	1	2		Msps

### 23.15.2 Radio current consumption (Transmitter)

Symbol	Description	Min.	Typ.	Max.	Units
$I_{TX,PLUS4dBm,DCDC}$	TX only run current (DCDC, 3V) $P_{RF} = +4$ dBm	7.5		mA	
$I_{TX,PLUS4dBm}$	TX only run current $P_{RF} = +4$ dBm	16.6		mA	
$I_{TX,0dBm,DCDC}$	TX only run current (DCDC, 3V) $P_{RF} = 0$ dBm	5.3		mA	
$I_{TX,0dBm}$	TX only run current $P_{RF} = 0$ dBm	11.6		mA	
$I_{TX,MINUS4dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -4$ dBm	4.2		mA	
$I_{TX,MINUS4dBm}$	TX only run current $P_{RF} = -4$ dBm	9.3		mA	
$I_{TX,MINUS8dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -8$ dBm	3.8		mA	
$I_{TX,MINUS8dBm}$	TX only run current $P_{RF} = -8$ dBm	8.4		mA	
$I_{TX,MINUS12dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -12$ dBm	3.5		mA	
$I_{TX,MINUS12dBm}$	TX only run current $P_{RF} = -12$ dBm	7.7		mA	
$I_{TX,MINUS16dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -16$ dBm	3.3		mA	
$I_{TX,MINUS16dBm}$	TX only run current $P_{RF} = -16$ dBm	7.3		mA	
$I_{TX,MINUS20dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -20$ dBm	3.2		mA	
$I_{TX,MINUS20dBm}$	TX only run current $P_{RF} = -20$ dBm	7.0		mA	
$I_{TX,MINUS40dBm,DCDC}$	TX only run current DCDC, 3V $P_{RF} = -40$ dBm	2.7		mA	

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>TX,MINUS40dBm</sub>	TX only run current P <sub>RF</sub> = -40 dBm	5.9			mA
I <sub>START,TX,DCDC</sub>	TX start-up current DCDC, 3V, P <sub>RF</sub> = 4 dBm	4.0			mA
I <sub>START,TX</sub>	TX start-up current, P <sub>RF</sub> = 4 dBm	8.8			mA

### 23.15.3 Radio current consumption (Receiver)

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>RX,1M,DCDC</sub>	RX only run current (DCDC, 3V) 1Msps / 1Msps BLE	5.4			mA
I <sub>RX,1M</sub>	RX only run current 1Msps / 1Msps BLE	11.7			mA
I <sub>RX,2M,DCDC</sub>	RX only run current (DCDC, 3V) 2Msps	5.8			mA
I <sub>RX,2M</sub>	RX only run current 2Msps	12.9			mA
I <sub>START,RX,1M,DCDC</sub>	RX start-up current (DCDC 3V) 1Msps / 1Msps BLE	3.5			mA
I <sub>START,RX,1M</sub>	RX start-up current 1Msps / 1Msps BLE	7.5			mA

### 23.15.4 Transmitter specification

Symbol	Description	Min.	Typ.	Max.	Units
P <sub>RF</sub>	Maximum output power	4	6		dBm
P <sub>RFC</sub>	RF power control range	24			dB
P <sub>RFCR</sub>	RF power accuracy			±4	dB
P <sub>RF1,1</sub>	1st Adjacent Channel Transmit Power 1 MHz (1 Msps)	-25			dBc
P <sub>RF2,1</sub>	2nd Adjacent Channel Transmit Power 2 MHz (1 Msps)	-50			dBc
P <sub>RF1,2</sub>	1st Adjacent Channel Transmit Power 2 MHz (2 Msps)	-25			dBc
P <sub>RF2,2</sub>	2nd Adjacent Channel Transmit Power 4 MHz (2 Msps)	-50			dBc

### 23.15.5 Receiver operation

Symbol	Description	Min.	Typ.	Max.	Units
P <sub>RX,MAX</sub>	Maximum received signal strength at < 0.1% PER	0			dBm
P <sub>SENS,IT,1M</sub>	Sensitivity, 1Msps nRF mode <sup>16</sup>	-93			dBm
P <sub>SENS,IT,SP,1M,BLE</sub>	Sensitivity, 1Msps BLE ideal transmitter, <=37 bytes BER=1E-3 <sup>17</sup>	-96			dBm
P <sub>SENS,IT,LP,1M,BLE</sub>	Sensitivity, 1Msps BLE ideal transmitter >=128 bytes BER=1E-4 <sup>18</sup>	-95			dBm
P <sub>SENS,IT,2M</sub>	Sensitivity, 2Msps nRF mode <sup>19</sup>	-89			dBm

### 23.15.6 RX selectivity

RX selectivity with equal modulation on interfering signal<sup>20</sup>

Symbol	Description	Min.	Typ.	Max.	Units
C/I <sub>1M,co-channel</sub>	1Msps mode, Co-Channel interference	9			dB
C/I <sub>1M,-1MHz</sub>	1 Msps mode, Adjacent (-1 MHz) interference	-2			dB
C/I <sub>1M,+1MHz</sub>	1 Msps mode, Adjacent (+1 MHz) interference	-10			dB
C/I <sub>1M,-2MHz</sub>	1 Msps mode, Adjacent (-2 MHz) interference	-19			dB
C/I <sub>1M,+2MHz</sub>	1 Msps mode, Adjacent (+2 MHz) interference	-42			dB
C/I <sub>1M,-3MHz</sub>	1 Msps mode, Adjacent (-3 MHz) interference	-38			dB
C/I <sub>1M,+3MHz</sub>	1 Msps mode, Adjacent (+3 MHz) interference	-48			dB
C/I <sub>1M,±6MHz</sub>	1 Msps mode, Adjacent (≥6 MHz) interference	-50			dB
C/I <sub>1MBLE,co-channel</sub>	1 Msps BLE mode, Co-Channel interference	6			dB

<sup>16</sup> Typical sensitivity applies when ADDR0 is used for receiver address correlation. When ADDR[1...7] are used for receiver address correlation, the typical sensitivity for this mode is degraded by 3dB.

<sup>17</sup> As defined in the Bluetooth Core Specification v4.0 Volume 6: Core System Package (Low Energy Controller Volume)

<sup>18</sup> Equivalent BER limit < 10E-04

<sup>19</sup> Typical sensitivity applies when ADDR0 is used for receiver address correlation. When ADDR[1...7] are used for receiver address correlation, the typical sensitivity for this mode is degraded by 3dB.

<sup>20</sup> Wanted signal level at PIN = -67 dBm. One interferer is used, having equal modulation as the wanted signal. The input power of the interferer where the sensitivity equals BER = 0.1% is presented

Symbol	Description	Min.	Typ.	Max.	Units
C/I <sub>1MBLE,-1MHz</sub>	1 Msps BLE mode, Adjacent (-1 MHz) interference	-2			dB
C/I <sub>1MBLE,+1MHz</sub>	1 Msps BLE mode, Adjacent (+1 MHz) interference	-9			dB
C/I <sub>1MBLE,-2MHz</sub>	1 Msps BLE mode, Adjacent (-2 MHz) interference	-22			dB
C/I <sub>1MBLE,+2MHz</sub>	1 Msps BLE mode, Adjacent (+2 MHz) interference	-46			dB
C/I <sub>1MBLE,&gt;3MHz</sub>	1 Msps BLE mode, Adjacent ( $\geq 3$ MHz) interference	-50			dB
C/I <sub>1MBLE,image</sub>	Image frequency Interference	-22			dB
C/I <sub>1MBLE,image,1MHz</sub>	Adjacent (1 MHz) interference to in-band image frequency	-35			dB
C/I <sub>2M,co-channel</sub>	2Msps mode, Co-Channel interference	10			dB
C/I <sub>2M,-2MHz</sub>	2 Msps mode, Adjacent (-2 MHz) interference	6			dB
C/I <sub>2M,+2MHz</sub>	2 Msps mode, Adjacent (+2 MHz) interference	-14			dB
C/I <sub>2M,-4MHz</sub>	2 Msps mode, Adjacent (-4 MHz) interference	-20			dB
C/I <sub>2M,+4MHz</sub>	2 Msps mode, Adjacent (+4 MHz) interference	-44			dB
C/I <sub>2M,-6MHz</sub>	2 Msps mode, Adjacent (-6 MHz) interference	-42			dB
C/I <sub>2M,+6MHz</sub>	2 Msps mode, Adjacent (+6 MHz) interference	-47			dB
C/I <sub>2M,&gt;12MHz</sub>	2 Msps mode, Adjacent ( $\geq 12$ MHz) interference	-52			dB

### 23.15.7 RX intermodulation

RX intermodulation<sup>21</sup>

Symbol	Description	Min.	Typ.	Max.	Units
P <sub>IMD,1M</sub>	IMD performance, 1 Msps, 3rd, 4th, and 5th offset channel	-33			dBm
P <sub>IMD,1M,BLE</sub>	IMD performance, BLE 1 Msps, 3rd, 4th, and 5th offset channel	-30			dBm
P <sub>IMD,2M</sub>	IMD performance, 2 Msps, 3rd, 4th, and 5th offset channel	-33			dBm

### 23.15.8 Radio timing

Symbol	Description	Min.	Typ.	Max.	Units
t <sub>TXEN</sub>	Time between TXEN task and READY event after channel FREQUENCY configured	140			us
t <sub>TXEN,FAST</sub>	Time between TXEN task and READY event after channel FREQUENCY configured (Fast Mode)	40			us
t <sub>TXDISABLE</sub>	Time between DISABLE task and DISABLED event when the radio was in TX and mode is set to 1Msps	6			us
t <sub>TXDISABLE,2M</sub>	Time between DISABLE task and DISABLED event when the radio was in TX and mode is set to 2Msps	4			us
t <sub>RXEN</sub>	Time between the RXEN task and READY event after channel FREQUENCY configured in default mode	140			us
t <sub>RXEN,FAST</sub>	Time between the RXEN task and READY event after channel FREQUENCY configured in fast mode	40			us
t <sub>SWITCH</sub>	The minimum time taken to switch from RX to TX or TX to RX (channel FREQUENCY unchanged)	20			us
t <sub>RXDISABLE</sub>	Time between DISABLE task and DISABLED event when the radio was in RX	0			us
t <sub>TXCHAIN</sub>	TX chain delay	0.6			us
t <sub>RXCHAIN</sub>	RX chain delay	9.4			us
t <sub>RXCHAIN,2M</sub>	RX chain delay in 2Msps mode	5			us

### 23.15.9 Received Signal Strength Indicator (RSSI) specifications

Symbol	Description	Min.	Typ.	Max.	Units
RSSI <sub>ACC</sub>	RSSI Accuracy Valid range -90 to -20 dBm	$\pm 2$			dB
RSSI <sub>RESOLUTION</sub>	RSSI resolution	1			dB
RSSI <sub>PERIOD</sub>	Sample period	8			us

<sup>21</sup> Wanted signal level at PIN = -64 dBm. Two interferers with equal input power are used. The interferer closest in frequency is not modulated, the other interferer is modulated equal with the wanted signal. The input power of the interferers where the sensitivity equals BER = 0.1% is presented.

### 23.15.10 Jitter

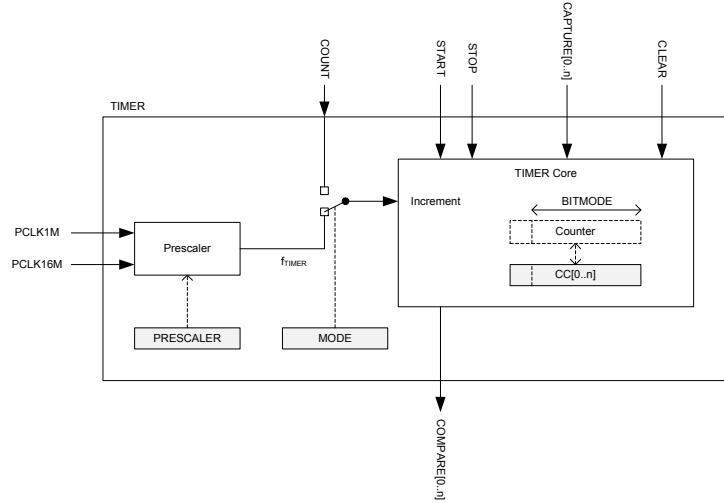
Symbol	Description	Min.	Typ.	Max.	Units
$t_{DISABLEDJITTER}$	Jitter on DISABLED event relative to END event when shortcut between END and DISABLE is enabled.		0.25		us
$t_{READYJITTER}$	Jitter on READY event relative to TXEN and RXEN task.		0.25		us

### 23.15.11 Delay when disabling the RADIO

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TXDISABLE,1M}$	Disable delay from TX.  Delay between DISABLE and DISABLED for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		6		us
$t_{RXDISABLE,1M}$	Disable delay from RX.  Delay between DISABLE and DISABLED for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		0		us

## 24 TIMER — Timer/counter

The TIMER can operate in two modes: timer and counter.



**Figure 42: Block schematic for timer/counter**

The timer/counter runs on the high-frequency clock source (HFCLK) and includes a four-bit (1/2X) prescaler that can divide the timer input clock from the HFCLK controller. Clock source selection between PCLK16M and PCLK1M is automatic according to TIMER base frequency set by the prescaler. The TIMER base frequency is always given as 16 MHz divided by the prescaler value.

The PPI system allows a TIMER event to trigger a task of any other system peripheral of the device. The PPI system also enables the TIMER task/event features to generate periodic output and PWM signals to any GPIO. The number of input/outputs used at the same time is limited by the number of GPIOTE channels.

The TIMER can operate in two modes, Timer mode and Counter mode. In both modes, the TIMER is started by triggering the START task, and stopped by triggering the STOP task. After the timer is stopped the timer can resume timing/counting by triggering the START task again. When timing/counting is resumed, the timer will continue from the value it had prior to being stopped.

In Timer mode, the TIMER's internal Counter register is incremented by one for every tick of the timer frequency  $f_{\text{TIMER}}$  as illustrated in [Figure 42: Block schematic for timer/counter](#) on page 233. The timer frequency is derived from PCLK16M as shown below, using the values specified in the PRESCALER register:

$$f_{\text{TIMER}} = 16 \text{ MHz} / (2^{\text{PRESCALER}})$$

When  $f_{\text{TIMER}} \leq 1 \text{ MHz}$  the TIMER will use PCLK1M instead of PCLK16M for reduced power consumption.

In counter mode, the TIMER's internal Counter register is incremented by one each time the COUNT task is triggered, that is, the timer frequency and the prescaler are not utilized in counter mode. Similarly, the COUNT task has no effect in Timer mode.

The TIMER's maximum value is configured by changing the bit-width of the timer in the [BITMODE](#) on page 238 register.

[PRESCALER](#) on page 238 and the [BITMODE](#) on page 238 must only be updated when the timer is stopped. If these registers are updated while the TIMER is started then this may result in unpredictable behavior.

When the timer is incremented beyond its maximum value the Counter register will overflow and the TIMER will automatically start over from zero.

The Counter register can be cleared, that is, its internal value set to zero explicitly, by triggering the CLEAR task.

The TIMER implements multiple capture/compare registers.

Independent of prescaler setting the accuracy of the TIMER is equivalent to one tick of the timer frequency  $f_{\text{TIMER}}$  as illustrated in [Figure 42: Block schematic for timer/counter](#) on page 233.

## 24.1 Capture

The TIMER implements one capture task for every available capture/compare register.

Every time the CAPTURE[n] task is triggered, the Counter value is copied to the CC[n] register.

## 24.2 Compare

The TIMER implements one COMPARE event for every available capture/compare register.

A COMPARE event is generated when the Counter is incremented and then becomes equal to the value specified in one of the capture compare registers. When the Counter value becomes equal to the value specified in a capture compare register CC[n], the corresponding compare event COMPARE[n] is generated.

[BITMODE](#) on page 238 specifies how many bits of the Counter register and the capture/compare register that are used when the comparison is performed. Other bits will be ignored.

## 24.3 Task delays

After the TIMER is started, the CLEAR task, COUNT task and the STOP task will guarantee to take effect within one clock cycle of the PCLK16M.

## 24.4 Task priority

If the START task and the STOP task are triggered at the same time, that is, within the same period of PCLK16M, the STOP task will be prioritized.

## 24.5 Registers

**Table 42: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40008000	TIMER	TIMER0	Timer 0	This timer instance has 4 CC registers (CC[0..3])
0x40009000	TIMER	TIMER1	Timer 1	This timer instance has 4 CC registers (CC[0..3])
0x4000A000	TIMER	TIMER2	Timer 2	This timer instance has 4 CC registers (CC[0..3])
0x4001A000	TIMER	TIMER3	Timer 3	This timer instance has 6 CC registers (CC[0..5])
0x4001B000	TIMER	TIMER4	Timer 4	This timer instance has 6 CC registers (CC[0..5])

**Table 43: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start Timer

Register	Offset	Description
TASKS_STOP	0x004	Stop Timer
TASKS_COUNT	0x008	Increment Timer (Counter mode only)
TASKS_CLEAR	0x00C	Clear time
TASKS_SHUTDOWN	0x010	Shut down timer
		Deprecated
TASKS_CAPTURE[0]	0x040	Capture Timer value to CC[0] register
TASKS_CAPTURE[1]	0x044	Capture Timer value to CC[1] register
TASKS_CAPTURE[2]	0x048	Capture Timer value to CC[2] register
TASKS_CAPTURE[3]	0x04C	Capture Timer value to CC[3] register
TASKS_CAPTURE[4]	0x050	Capture Timer value to CC[4] register
TASKS_CAPTURE[5]	0x054	Capture Timer value to CC[5] register
EVENTS_COMPARE[0]	0x140	Compare event on CC[0] match
EVENTS_COMPARE[1]	0x144	Compare event on CC[1] match
EVENTS_COMPARE[2]	0x148	Compare event on CC[2] match
EVENTS_COMPARE[3]	0x14C	Compare event on CC[3] match
EVENTS_COMPARE[4]	0x150	Compare event on CC[4] match
EVENTS_COMPARE[5]	0x154	Compare event on CC[5] match
<b>SHORTS</b>	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
MODE	0x504	Timer mode selection
BITMODE	0x508	Configure the number of bits used by the TIMER
PRESCALER	0x510	Timer prescaler register
CC[0]	0x540	Capture/Compare register 0
CC[1]	0x544	Capture/Compare register 1
CC[2]	0x548	Capture/Compare register 2
CC[3]	0x54C	Capture/Compare register 3
CC[4]	0x550	Capture/Compare register 4
CC[5]	0x554	Capture/Compare register 5

## 24.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Id		L	K	J	I	H	G	F	E	D	C	B	A																																				
<b>Reset 0x00000000</b>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																																											
A	RW	COMPARE0_CLEAR				Shortcut between COMPARE[0] event and CLEAR task																																											
						See <a href="#">EVENTS_COMPARE[0]</a> and <a href="#">TASKS_CLEAR</a>																																											
						Disabled	0	Disable shortcut																																									
						Enabled	1	Enable shortcut																																									
B	RW	COMPARE1_CLEAR				Shortcut between COMPARE[1] event and CLEAR task																																											
						See <a href="#">EVENTS_COMPARE[1]</a> and <a href="#">TASKS_CLEAR</a>																																											
						Disabled	0	Disable shortcut																																									
						Enabled	1	Enable shortcut																																									
C	RW	COMPARE2_CLEAR				Shortcut between COMPARE[2] event and CLEAR task																																											
						See <a href="#">EVENTS_COMPARE[2]</a> and <a href="#">TASKS_CLEAR</a>																																											
						Disabled	0	Disable shortcut																																									
						Enabled	1	Enable shortcut																																									
D	RW	COMPARE3_CLEAR				Shortcut between COMPARE[3] event and CLEAR task																																											
						See <a href="#">EVENTS_COMPARE[3]</a> and <a href="#">TASKS_CLEAR</a>																																											
						Disabled	0	Disable shortcut																																									
						Enabled	1	Enable shortcut																																									
E	RW	COMPARE4_CLEAR				Shortcut between COMPARE[4] event and CLEAR task																																											
						See <a href="#">EVENTS_COMPARE[4]</a> and <a href="#">TASKS_CLEAR</a>																																											

### **24.5.2 INTENSET**

Address offset: 0x304

Enable interrupt

### 24.5.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Id			F	E	D	C	B	A																									
<b>Reset 0x00000000</b>			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>			<b>Value</b>			<b>Description</b>																								
A	RW	COMPARE0				Write '1' to Disable interrupt for COMPARE[0] event																											
						See <a href="#">EVENTS_COMPARE[0]</a>																											
			Clear			1			Disable																								
			Disabled			0			Read: Disabled																								
			Enabled			1			Read: Enabled																								
B	RW	COMPARE1				Write '1' to Disable interrupt for COMPARE[1] event																											
						See <a href="#">EVENTS_COMPARE[1]</a>																											
			Clear			1			Disable																								
			Disabled			0			Read: Disabled																								
			Enabled			1			Read: Enabled																								
C	RW	COMPARE2				Write '1' to Disable interrupt for COMPARE[2] event																											
						See <a href="#">EVENTS_COMPARE[2]</a>																											
			Clear			1			Disable																								
			Disabled			0			Read: Disabled																								
			Enabled			1			Read: Enabled																								
D	RW	COMPARE3				Write '1' to Disable interrupt for COMPARE[3] event																											
						See <a href="#">EVENTS_COMPARE[3]</a>																											
			Clear			1			Disable																								
			Disabled			0			Read: Disabled																								
			Enabled			1			Read: Enabled																								
E	RW	COMPARE4				Write '1' to Disable interrupt for COMPARE[4] event																											
						See <a href="#">EVENTS_COMPARE[4]</a>																											
			Clear			1			Disable																								
			Disabled			0			Read: Disabled																								
			Enabled			1			Read: Enabled																								

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F E D C B A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F RW COMPARES					Write '1' to Disable interrupt for COMPARE[5] event See <a href="#">EVENTS_COMPARE[5]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

## 24.5.4 MODE

Address offset: 0x504

Timer mode selection

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A RW MODE					Timer mode
			Timer	0	Select Timer mode
			Counter	1	Select Counter mode
			LowPowerCounter	2	Select Low Power Counter mode

## 24.5.5 BITMODE

Address offset: 0x508

Configure the number of bits used by the TIMER

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A RW BITMODE					Timer bit width
			16Bit	0	16 bit timer bit width
			08Bit	1	8 bit timer bit width
			24Bit	2	24 bit timer bit width
			32Bit	3	32 bit timer bit width

## 24.5.6 PRESCALER

Address offset: 0x510

Timer prescaler register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A				
Reset 0x00000004	0 0				
Id	RW	Field	Value Id	Value	Description
A RW PRESCALER				[0..9]	Prescaler value

## 24.5.7 CC[0]

Address offset: 0x540

Capture/Compare 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CC			Capture/Compare value																											
					Only the number of bits indicated by BITMODE will be used by the TIMER.																											

## 24.5.8 CC[1]

Address offset: 0x544

## Capture/Compare 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CC			Capture/Compare value																											
					Only the number of bits indicated by BITMODE will be used by the TIMER.																											

## 24.5.9 CC[2]

Address offset: 0x548

## Capture/Compare register 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CC			Capture/Compare value																											
					Only the number of bits indicated by BITMODE will be used by the TIMER.																											

## 24.5.10 CC[3]

Address offset: 0x54C

### Capture/Compare register 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CC			Capture/Compare value																											
					Only the number of bits indicated by BITMODE will be used by the TIMER.																											

## 24.5.11 CC[4]

Address offset: 0x550

## Capture/Compare register 4

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																												
A	RW	CC			Capture/Compare value																												

## 24.5.12 CC[5]

Address offset: 0x554

### Capture/Compare register 5

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																										
A	RW	CC				Capture/Compare value																										
						Only the number of bits indicated by BITMODE will be used by the TIMER.																										

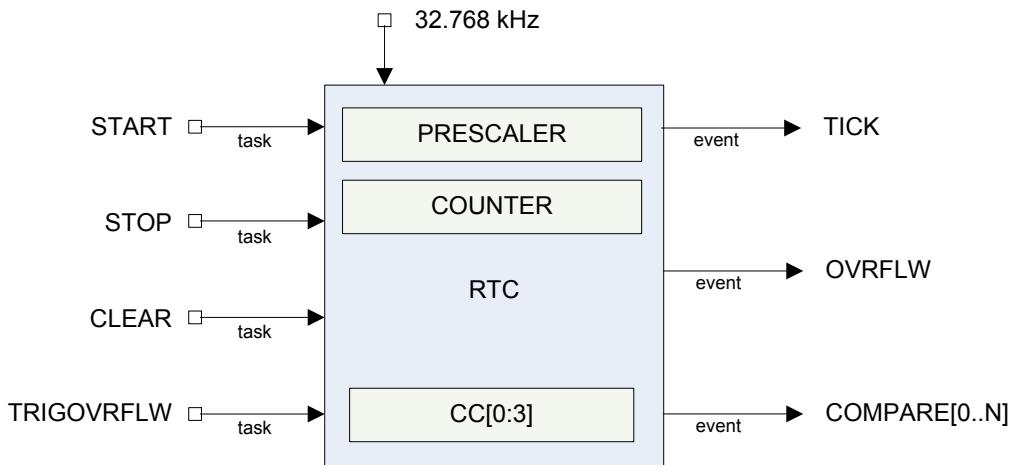
## **24.6 Electrical specification**

#### **24.6.1 Timers Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
I_TIMER_1M	Run current with 1 MHz clock input (PCLK1M)	3	5	8	µA
I_TIMER_16M	Run current with 16 MHz clock input (PCLK16M)	50	70	120	µA
t_TIMER,START	Time from START task is given until timer starts counting		0.25		µs

## 25 RTC — Real-time counter

The Real-time counter (RTC) module provides a generic, low power timer on the low-frequency clock source (LFCLK).



**Figure 43: RTC block schematic**

The RTC module features a 24-bit COUNTER, a 12-bit (1/X) prescaler, capture/compare registers, and a tick event generator for low power, tickless RTOS implementation.

### 25.1 Clock source

The RTC will run off the LFCLK.

The COUNTER resolution will therefore be 30.517 µs. Depending on the source, the RTC is able to run while the HFCLK is OFF and PCLK16M is not available.

The software has to explicitly start LFCLK before using the RTC.

See [CLOCK — Clock control](#) on page 101 for more information about clock sources.

### 25.2 Resolution versus overflow and the PRESCALER

Counter increment frequency:

$$f_{\text{RTC}} \text{ [kHz]} = 32.768 / (\text{PRESCALER} + 1)$$

The PRESCALER register is read/write when the RTC is stopped. The PRESCALER register is read-only once the RTC is STARTed. Writing to the PRESCALER register when the RTC is started has no effect.

The PRESCALER is restarted on START, CLEAR and TRIGOVRFLW, that is, the prescaler value is latched to an internal register (<<PRESC>>) on these tasks.

Examples:

- Desired COUNTER frequency 100 Hz (10 ms counter period)

$$\text{PRESCALER} = \text{round}(32.768 \text{ kHz} / 100 \text{ Hz}) - 1 = 327$$

$$f_{\text{RTC}} = 99.9 \text{ Hz}$$

10009.576  $\mu$ s counter period

2. Desired COUNTER frequency 8 Hz (125 ms counter period)

$$\text{PRESCALER} = \text{round}(32.768 \text{ kHz} / 8 \text{ Hz}) - 1 = 4095$$

$$f_{\text{RTC}} = 8 \text{ Hz}$$

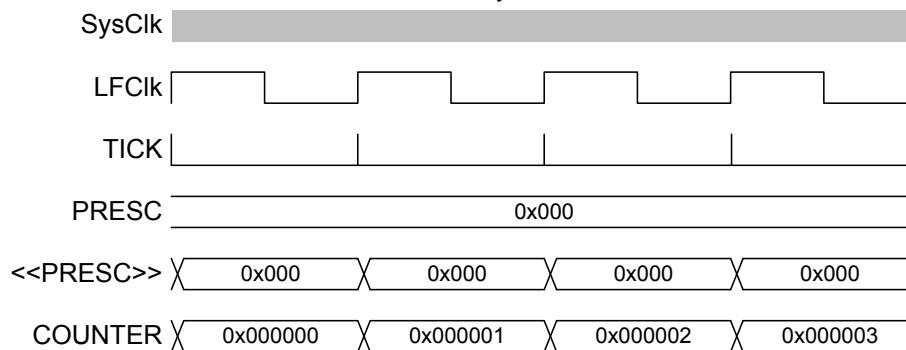
125 ms counter period

**Table 44: RTC resolution versus overflow**

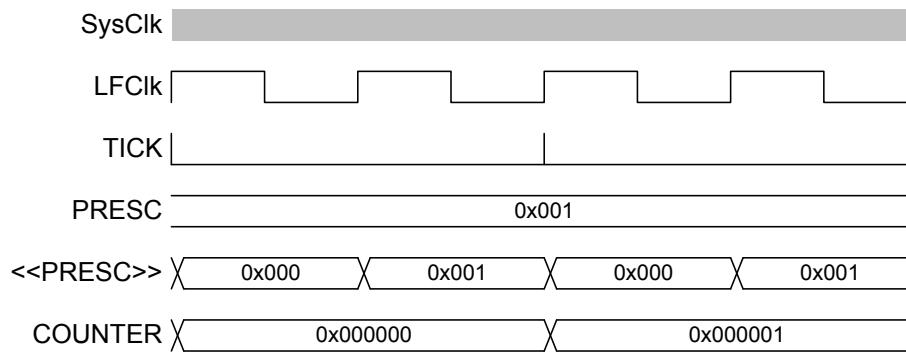
Prescaler	Counter resolution	Overflow
0	30.517 $\mu$ s	512 seconds
$2^8-1$	7812.5 $\mu$ s	131072 seconds
$2^{12}-1$	125 ms	582.542 hours

## 25.3 COUNTER register

The COUNTER increments on LFCLK when the internal PRESCALER register (<<PRESC>>) is 0x00. <<PRESC>> is reloaded from the PRESCALER register. If enabled, the TICK event occurs on each increment of the COUNTER. The TICK event is disabled by default.



**Figure 44: Timing diagram - COUNTER\_PRESCALER\_0**



**Figure 45: Timing diagram - COUNTER\_PRESCALER\_1**

## 25.4 Overflow features

The TRIGOVRFLW task sets the COUNTER value to 0xFFFFF0 to allow SW test of the overflow condition. OVRFLW occurs when COUNTER overflows from 0xFFFFFFF to 0.

**Important:** The OVRFLW event is disabled by default.

## 25.5 TICK event

The TICK event enables low power "tick-less" RTOS implementation as it optionally provides a regular interrupt source for a RTOS without the need to use the ARM® SysTick feature.

Using the RTC TICK event rather than the SysTick allows the CPU to be powered down while still keeping RTOS scheduling active.

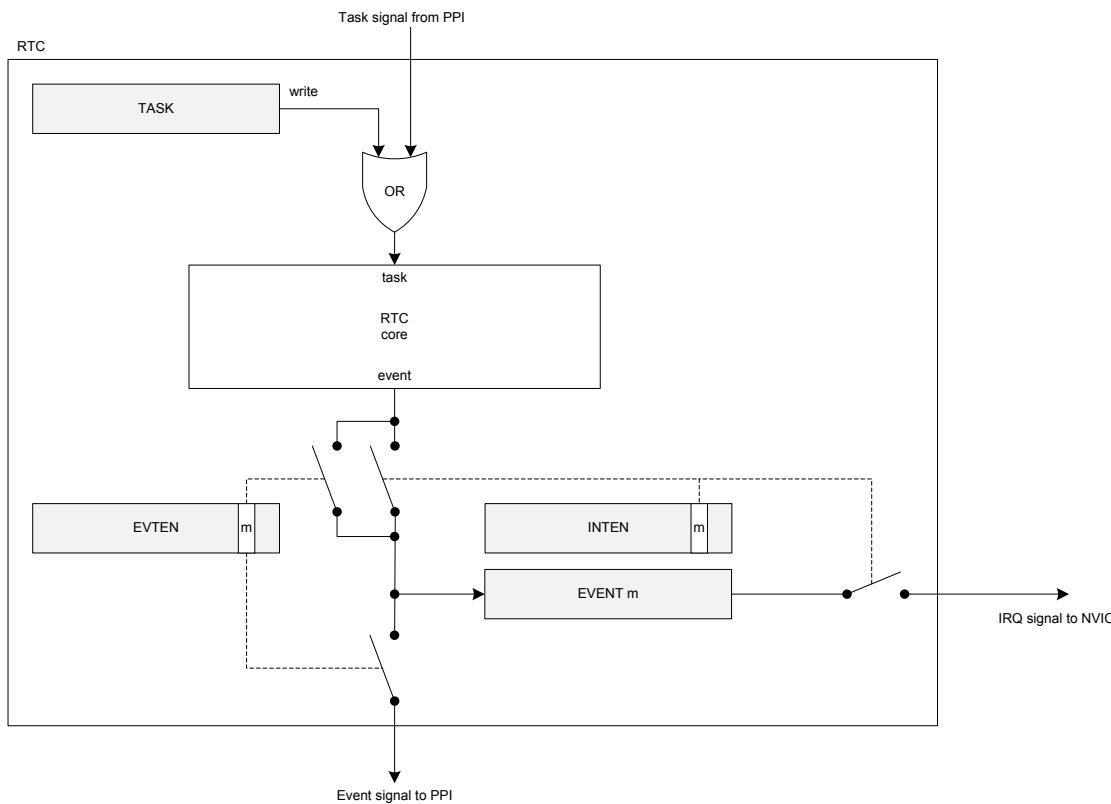
**Important:** The TICK event is disabled by default.

## 25.6 Event control feature

To optimize RTC power consumption, events in the RTC can be individually disabled to prevent PCLK16M and HFCLK being requested when those events are triggered. This is managed using the EVTEN register.

For example, if the TICK event is not required for an application, this event should be disabled as it is frequently occurring and may increase power consumption if HFCLK otherwise could be powered down for long durations.

This means that the RTC implements a slightly different task and event system compared to the standard system described in [Peripheral interface](#) on page 68. The RTC task and event system is illustrated in [Figure 46: Tasks, events and interrupts in the RTC](#) on page 243.



**Figure 46: Tasks, events and interrupts in the RTC**

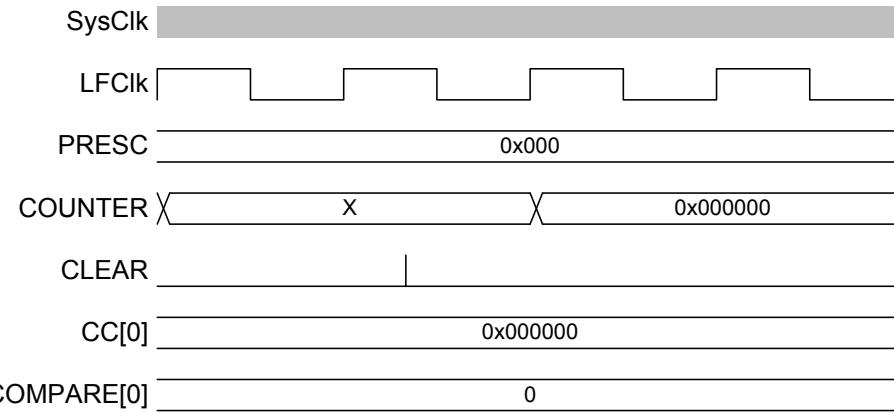
## 25.7 Compare feature

There are a number of Compare registers.

For more information, see [Registers](#) on page 247.

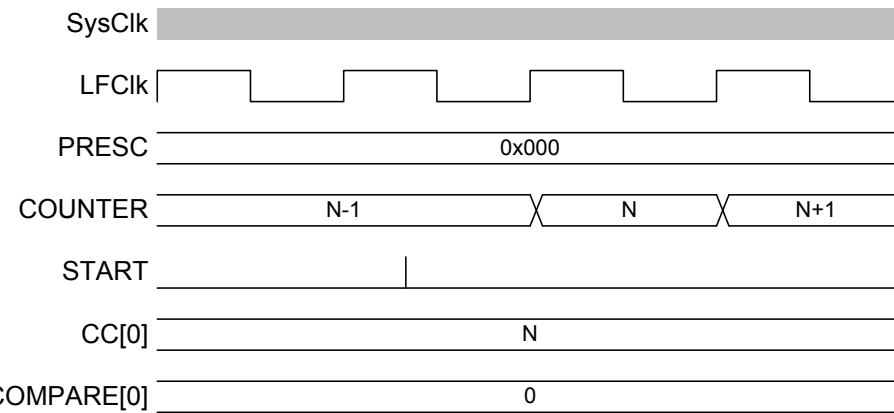
When setting a compare register, the following behavior of the RTC compare event should be noted:

- If a CC register value is 0 when a CLEAR task is set, this will not trigger a COMPARE event.



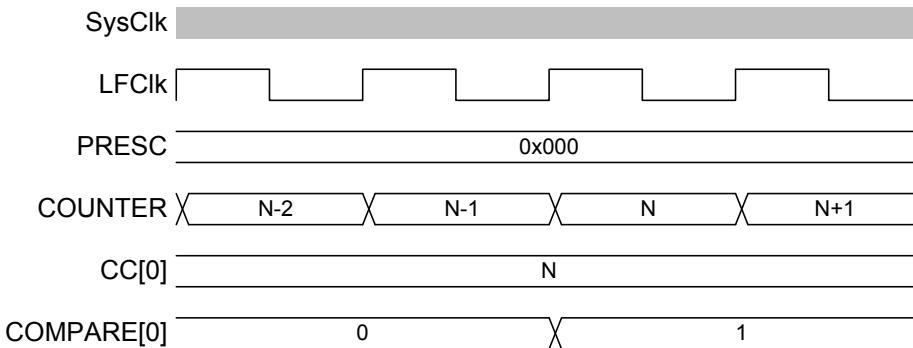
**Figure 47: Timing diagram - COMPARE\_CLEAR**

- If a CC register is N and the COUNTER value is N when the START task is set, this will not trigger a COMPARE event.



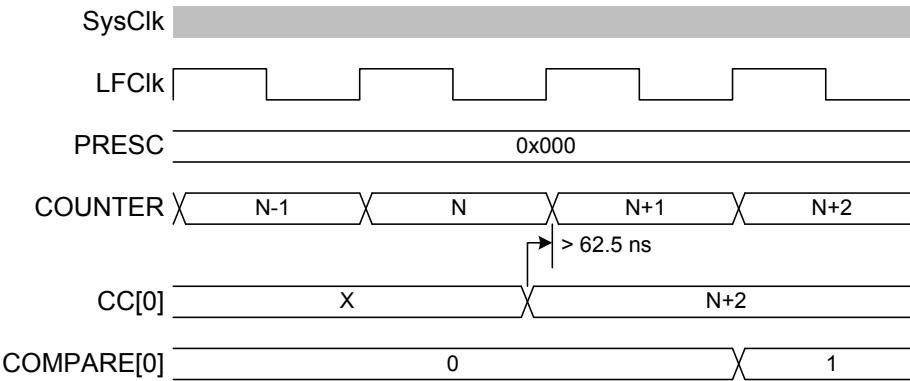
**Figure 48: Timing diagram - COMPARE\_START**

- COMPARE occurs when a CC register is N and the COUNTER value transitions from N-1 to N.



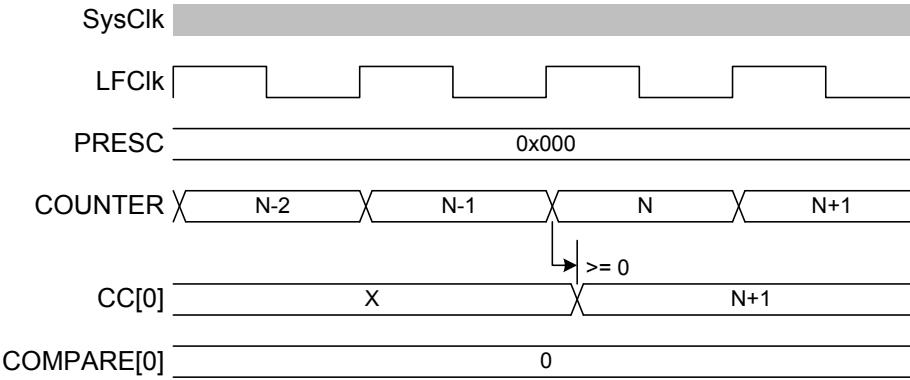
**Figure 49: Timing diagram - COMPARE**

- If the COUNTER is N, writing N+2 to a CC register is guaranteed to trigger a COMPARE event at N+2.



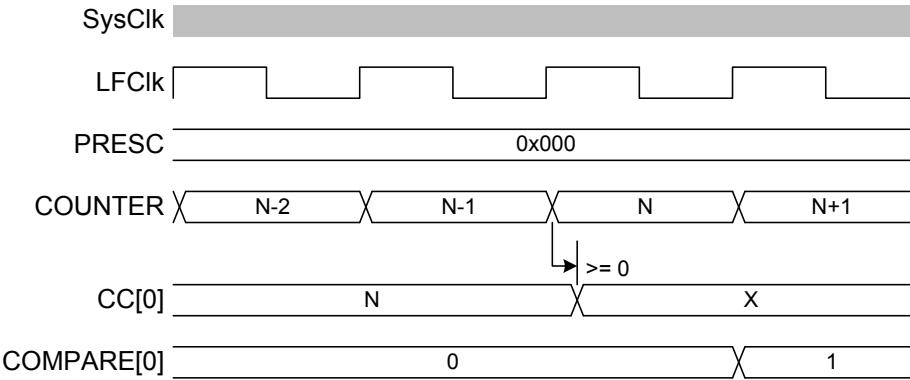
**Figure 50: Timing diagram - COMPARE\_N+2**

- If the COUNTER is N, writing N or N+1 to a CC register may not trigger a COMPARE event.



**Figure 51: Timing diagram - COMPARE\_N+1**

- If the COUNTER is N and the current CC register value is N+1 or N+2 when a new CC value is written, a match may trigger on the previous CC value before the new value takes effect. If the current CC value is greater than N+2 when the new value is written, there will be no event due to the old value.



**Figure 52: Timing diagram - COMPARE\_N-1**

## 25.8 TASK and EVENT jitter/delay

Jitter or delay in the RTC is due to the peripheral clock being a low frequency clock (LFCLK) which is not synchronous to the faster PCLK16M.

Registers in the peripheral interface, part of the PCLK16M domain, have a set of mirrored registers in the LFCLK domain. For example, the COUNTER value accessible from the CPU is in the PCLK16M domain and is latched on read from an internal register called COUNTER in the LFCLK domain. COUNTER is the register which is actually modified each time the RTC ticks. These registers must be synchronised between clock domains (PCLK16M and LFCLK).

The following is a summary of the jitter introduced on tasks and events. Figures illustrating jitter follow.

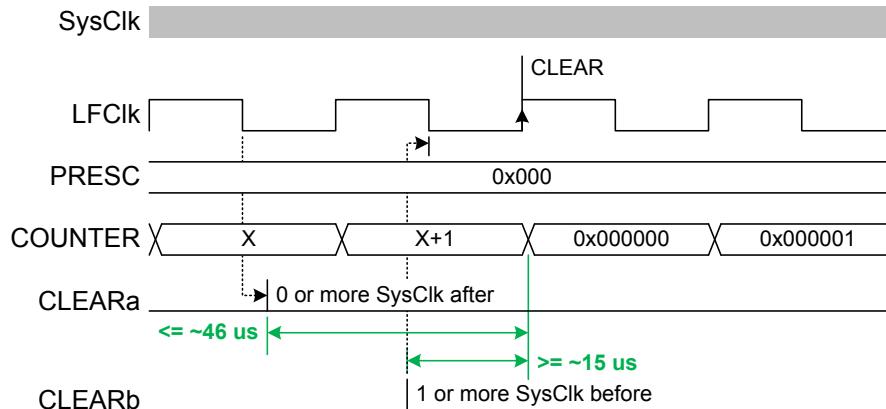
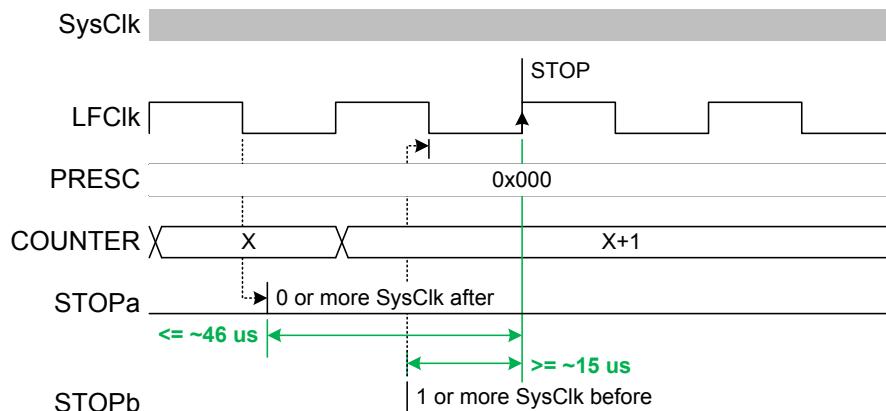
**Table 45: RTC jitter magnitudes on tasks**

Task	Delay
CLEAR, STOP, START, TRIGOVRFLW	+15 to 46 $\mu$ s

**Table 46: RTC jitter magnitudes on events**

Operation/Function	Jitter
START to COUNTER increment	+/- 15 $\mu$ s
COMPARE to COMPARE <sup>22</sup>	+/- 62.5 ns

1. CLEAR and STOP (and TRIGOVRFLW; not shown) will be delayed as long as it takes for the peripheral to clock a falling edge and rising of the LFCLK. This is between 15.2585  $\mu$ s and 45.7755  $\mu$ s – rounded to 15  $\mu$ s and 46  $\mu$ s for the remainder of the section.


**Figure 53: Timing diagram - DELAY\_CLEAR**

**Figure 54: Timing diagram - DELAY\_STOP**

2. The START task will start the RTC. Assuming that the LFCLK was previously running and stable, the first increment of COUNTER (and instance of TICK event) will be typically after 30.5  $\mu$ s +/-15  $\mu$ s. In some cases, in particular if the RTC is STARTed before the LFCLK is running, that timing can be up to ~250  $\mu$ s. The software should therefore wait for the first TICK if it has to make sure the RTC is running. Sending a TRIGOVRFLW task sets the COUNTER to a value close to overflow. However, since the update of COUNTER relies on a stable LFCLK, sending this task while LFCLK is not running will start LFCLK, but the update will then be delayed by the same amount of time of up to ~250  $\mu$ s. The figures show the smallest and largest delays to on the START task which appears as a +/-15  $\mu$ s jitter on the first COUNTER increment.

<sup>22</sup> Assumes RTC runs continuously between these events.

**Note:** 32.768 kHz clock jitter is additional to the numbers provided above.

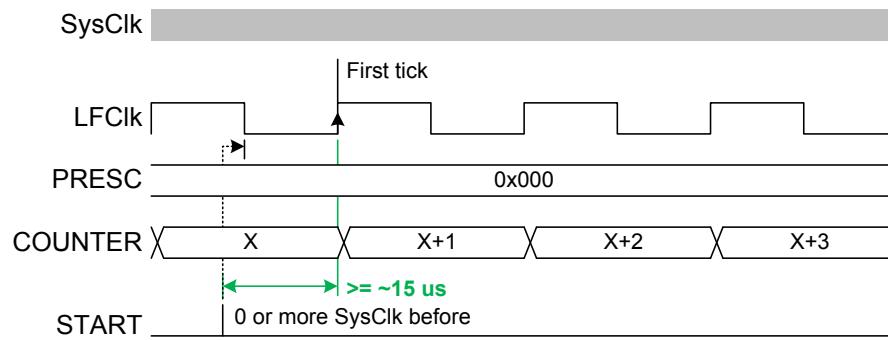


Figure 55: Timing diagram - JITTER\_START-

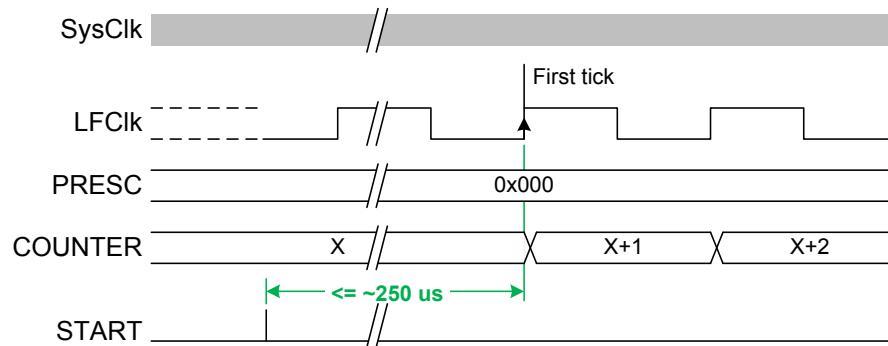


Figure 56: Timing diagram - JITTER\_START+

## 25.9 Reading the COUNTER register

To read the COUNTER register, the internal <<COUNTER>> value is sampled.

To ensure that the <<COUNTER>> is safely sampled (considering an LFCLK transition may occur during a read), the CPU and core memory bus are halted for three cycles by lowering the core PREADY signal. The Read takes the CPU 2 cycles in addition resulting in the COUNTER register read taking a fixed five PCLK16M clock cycles.

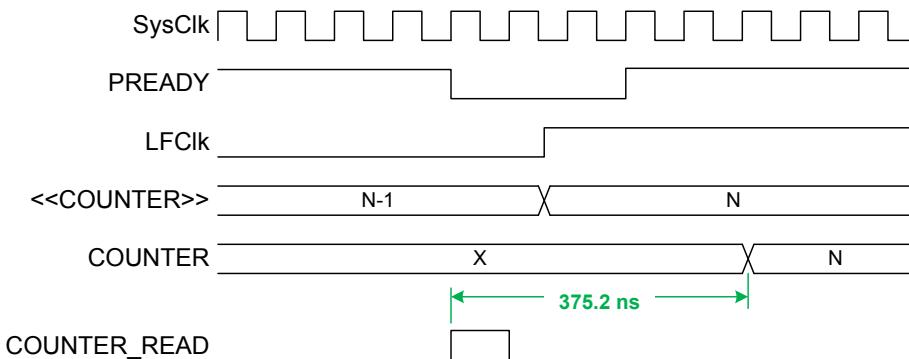


Figure 57: Timing diagram - COUNTER\_READ

## 25.10 Registers

Table 47: Instances

Base address	Peripheral	Instance	Description	Configuration
0x4000B000	RTC	RTC0	Real-time counter 0	CC[0..2] implemented, CC[3] not implemented
0x40011000	RTC	RTC1	Real-time counter 1	CC[0..3] implemented

Base address	Peripheral	Instance	Description	Configuration
0x40024000	RTC	RTC2	Real-time counter 2	CC[0..3] implemented

**Table 48: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start RTC COUNTER
TASKS_STOP	0x004	Stop RTC COUNTER
TASKS_CLEAR	0x008	Clear RTC COUNTER
TASKS_TRIGOVRLW	0x00C	Set COUNTER to 0xFFFFF0
EVENTS_TICK	0x100	Event on COUNTER increment
EVENTS_OVRFLOW	0x104	Event on COUNTER overflow
EVENTS_COMPARE[0]	0x140	Compare event on CC[0] match
EVENTS_COMPARE[1]	0x144	Compare event on CC[1] match
EVENTS_COMPARE[2]	0x148	Compare event on CC[2] match
EVENTS_COMPARE[3]	0x14C	Compare event on CC[3] match
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>EVTEN</i>	0x340	Enable or disable event routing
<i>EVTENSET</i>	0x344	Enable event routing
<i>EVTENCLR</i>	0x348	Disable event routing
<i>COUNTER</i>	0x504	Current COUNTER value
<i>PRESCALER</i>	0x508	12 bit prescaler for COUNTER frequency (32768/(PRESCALER+1)).Must be written when RTC is stopped
<i>CC[0]</i>	0x540	Compare register 0
<i>CC[1]</i>	0x544	Compare register 1
<i>CC[2]</i>	0x548	Compare register 2
<i>CC[3]</i>	0x54C	Compare register 3

## 25.10.1 INTENSET

Address offset: 0x304

## Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F E D C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
E	RW	COMPARE2			Write '1' to Enable interrupt for COMPARE[2] event  See <a href="#">EVENTS_COMPARE[2]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to Enable interrupt for COMPARE[3] event  See <a href="#">EVENTS_COMPARE[3]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

## 25.10.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F E D C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	TICK			Write '1' to Disable interrupt for TICK event  See <a href="#">EVENTS_TICK</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to Disable interrupt for OVRFLW event  See <a href="#">EVENTS_OVRFLW</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	COMPARE0			Write '1' to Disable interrupt for COMPARE[0] event  See <a href="#">EVENTS_COMPARE[0]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	COMPARE1			Write '1' to Disable interrupt for COMPARE[1] event  See <a href="#">EVENTS_COMPARE[1]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	COMPARE2			Write '1' to Disable interrupt for COMPARE[2] event  See <a href="#">EVENTS_COMPARE[2]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	COMPARE3			Write '1' to Disable interrupt for COMPARE[3] event  See <a href="#">EVENTS_COMPARE[3]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

### 25.10.3 EVTEN

Address offset: 0x340

Enable or disable event routing

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				F	E	D	C	B	A																										
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Id	RW	Field	Value Id	Value				Description																											
A	RW	TICK		Enable or disable event routing for TICK event																						See <a href="#">EVENTS_TICK</a>									
				Disabled	0	Disable																													
				Enabled	1	Enable																													
B	RW	OVRFLOW		Enable or disable event routing for OVRFLOW event																							See <a href="#">EVENTS_OVRFLOW</a>								
				Disabled	0	Disable																													
				Enabled	1	Enable																													
C	RW	COMPARE0		Enable or disable event routing for COMPARE[0] event																							See <a href="#">EVENTS_COMPARE[0]</a>								
				Disabled	0	Disable																													
				Enabled	1	Enable																													
D	RW	COMPARE1		Enable or disable event routing for COMPARE[1] event																							See <a href="#">EVENTS_COMPARE[1]</a>								
				Disabled	0	Disable																													
				Enabled	1	Enable																													
E	RW	COMPARE2		Enable or disable event routing for COMPARE[2] event																							See <a href="#">EVENTS_COMPARE[2]</a>								
				Disabled	0	Disable																													
				Enabled	1	Enable																													
F	RW	COMPARE3		Enable or disable event routing for COMPARE[3] event																							See <a href="#">EVENTS_COMPARE[3]</a>								
				Disabled	0	Disable																													
				Enabled	1	Enable																													

### 25.10.4 EVTENSET

Address offset: 0x344

Enable event routing

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				F	E	D	C	B	A																										
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Id	RW	Field	Value Id	Value				Description																											
A	RW	TICK		Write '1' to Enable event routing for TICK event																							See <a href="#">EVENTS_TICK</a>								
				Set	1	Enable																													
				Disabled	0	Read: Disabled																													
				Enabled	1	Read: Enabled																													
B	RW	OVRFLOW		Write '1' to Enable event routing for OVRFLOW event																							See <a href="#">EVENTS_OVRFLOW</a>								
				Set	1	Enable																													
				Disabled	0	Read: Disabled																													
				Enabled	1	Read: Enabled																													
C	RW	COMPARE0		Write '1' to Enable event routing for COMPARE[0] event																							See <a href="#">EVENTS_COMPARE[0]</a>								
				Set	1	Enable																													

## 25.10.5 EVTENCLR

Address offset: 0x348

## Disable event routing

## 25.10.6 COUNTER

Address offset: 0x504

Current COUNTER value

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	COUNTER			Counter value																											

## 25.10.7 PRESCALER

Address offset: 0x508

12 bit prescaler for COUNTER frequency ( $32768 / (\text{PRESCALER} + 1)$ ). Must be written when RTC is stopped

## 25.10.8 CC[0]

Address offset: 0x540

## Compare register 0

## 25.10.9 CC[1]

Address offset: 0x544

Compare register 1

25.10.10 CC[2]

Address offset: 0x548

Compare register 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	COMPARE			Compare value																											

25.10.11 CC[3]

Address offset: 0x54C

### Compare register 3

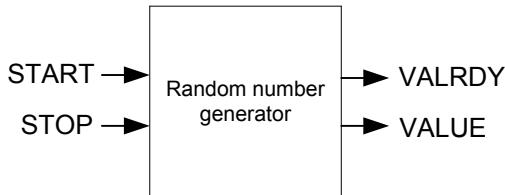
## **25.11 Electrical specification**

### **25.11.1 RTC Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>RTC</sub>	Run current Real Time Counter (LFCLK source)		0.1		µA

# 26 RNG — Random number generator

The Random number generator (RNG) generates true non-deterministic random numbers based on internal thermal noise that are suitable for cryptographic purposes. The RNG does not require a seed value.



**Figure 58: Random number generator**

The RNG is started by triggering the START task and stopped by triggering the STOP task. When started, new random numbers are generated continuously and written to the VALUE register when ready. A VALRDY event is generated for every new random number that is written to the VALUE register. This means that after a VALRDY event is generated the CPU has the time until the next VALRDY event to read out the random number from the VALUE register before it is overwritten by a new random number.

## 26.1 Bias correction

A bias correction algorithm is employed on the internal bit stream to remove any bias toward '1' or '0'. The bits are then queued into an eight-bit register for parallel readout from the VALUE register.

It is possible to enable bias correction in the CONFIG register. This will result in slower value generation, but will ensure a statistically uniform distribution of the random values.

## 26.2 Speed

The time needed to generate one random byte of data is unpredictable, and may vary from one byte to the next. This is especially true when bias correction is enabled.

## 26.3 Registers

**Table 49: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4000D000	RNG	RNG	Random number generator	

**Table 50: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Task starting the random number generator
TASKS_STOP	0x004	Task stopping the random number generator
EVENTS_VALRDY	0x100	Event being generated for every new random number written to the VALUE register
<b>SHORTS</b>	0x200	Shortcut register
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>CONFIG</b>	0x504	Configuration register
<b>VALUE</b>	0x508	Output random number

### 26.3.1 SHORTS

Address offset: 0x200

Shortcut register

### **26.3.2 INTENSET**

Address offset: 0x304

## Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	VALRDY			Write '1' to Enable interrupt for VALRDY event																											
					See <a href="#">EVENTS_VALRDY</a>																											
			Set	1	Enable																											
			Disabled	0	Read: Disabled																											
			Enabled	1	Read: Enabled																											

### 26.3.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	VALRDY		Write '1' to Disable interrupt for VALRDY event																												
				See <a href="#">EVENTS_VALRDY</a>																												
			Clear	1	Disable																											
			Disabled	0	Read: Disabled																											
			Enabled	1	Read: Enabled																											

### **26.3.4 CONFIG**

Address offset: 0x504

## Configuration register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id	A																																
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value				Description																									
A	RW	DERCEN		Disabled				Bias correction																									
			Enabled	0				Disabled																									
			Enabled	1				Enabled																									

### **26.3.5 VALUE**

Address offset: 0x508

## Output random number

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	R	VALUE		[0..255]	Generated random number

## 26.4 Electrical specification

### 26.4.1 RNG Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$I_{RNG}$	Run current, CPU sleeping.	500			$\mu A$
$t_{RNG,START}$	Time from setting the START task to generation begins. This is a one-time delay on START signal and does not apply between samples.	128			$\mu s$
$t_{RNG,RAW}$	Run time per byte without bias correction. Uniform distribution of 0 and 1 is not guaranteed.	30			$\mu s$
$t_{RNG,BC}$	Run time per byte with bias correction. Uniform distribution of 0 and 1 is guaranteed. Time to generate a byte cannot be guaranteed.	120			$\mu s$

# 27 TEMP — Temperature sensor

The temperature sensor measures die temperature over the temperature range of the device. Linearity compensation can be implemented if required by the application.

Listed here are the main features for TEMP:

- Temperature range is greater than or equal to operating temperature of the device
- Resolution is 0.25 degrees

TEMP is started by triggering the START task.

When the temperature measurement is completed, a DATARDY event will be generated and the result of the measurement can be read from the TEMP register.

To achieve the measurement accuracy stated in the electrical specification, the crystal oscillator must be selected as the HFCLK source, see [CLOCK — Clock control](#) on page 101 for more information.

When the temperature measurement is completed, TEMP analog electronics power down to save power.

TEMP only supports one-shot operation, meaning that every TEMP measurement has to be explicitly started using the START task.

## 27.1 Registers

**Table 51: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4000C000	TEMP	TEMP	Temperature sensor	

**Table 52: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start temperature measurement
TASKS_STOP	0x004	Stop temperature measurement
EVENTS_DATARDY	0x100	Temperature measurement complete, data ready
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
TEMP	0x508	Temperature in °C (0.25° steps)
A0	0x520	Slope of 1st piece wise linear function
A1	0x524	Slope of 2nd piece wise linear function
A2	0x528	Slope of 3rd piece wise linear function
A3	0x52C	Slope of 4th piece wise linear function
A4	0x530	Slope of 5th piece wise linear function
A5	0x534	Slope of 6th piece wise linear function
B0	0x540	y-intercept of 1st piece wise linear function
B1	0x544	y-intercept of 2nd piece wise linear function
B2	0x548	y-intercept of 3rd piece wise linear function
B3	0x54C	y-intercept of 4th piece wise linear function
B4	0x550	y-intercept of 5th piece wise linear function
B5	0x554	y-intercept of 6th piece wise linear function
T0	0x560	End point of 1st piece wise linear function
T1	0x564	End point of 2nd piece wise linear function
T2	0x568	End point of 3rd piece wise linear function
T3	0x56C	End point of 4th piece wise linear function
T4	0x570	End point of 5th piece wise linear function

## 27.1.1 INTENSET

Address offset: 0x304

## Enable interrupt

## 27.1.2 INTENCLR

Address offset: 0x308

Disable interrupt

### 27.1.3 TEMP

Address offset: 0x508

### Temperature in °C (0.25° steps)

## 27.1.4 A0

Address offset: 0x520

## Slope of 1st piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000320</b>	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	
Id	RW	Field	Value Id	Value	Description																											
A	RW	A0			Slope of 1st piece wise linear function																											

27.1.5 A1

Address offset: 0x524

## Slope of 2nd piece wise linear function

## 27.1.6 A2

Address offset: 0x528

Slope of 3rd piece wise linear function

### 27.1.7 A3

Address offset: 0x52C

## Slope of 4th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000400</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value																		Description										
A	RW	A3																				Slope of 4th piece wise linear function										

27.1.8 A4

Address offset: 0x530

## Slope of 5th piece wise linear function

27.1.9 A5

Address offset: 0x534

Slope of 6th piece wise linear function

27.1.10 B0

Address offset: 0x540

y-intercept of 1st piece wise linear function

27.1.11 B1

Address offset: 0x544

y-intercept of 2nd piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Id																			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00003F98</b>	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	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																																	
A	RW	B1			y-intercept of 2nd piece wise linear function																																	

27.1.12 B2

Address offset: 0x548

y-intercept of 3rd piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Id																			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0x00003F98</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	0	0	0						
Id	RW	Field	Value Id	Value	Description																																	
A	RW	B2			y-intercept of 3rd piece wise linear function																																	

27.1.13 B3

Address offset: 0x54C

y-intercept of 4th piece wise linear function

27.1.14 B4

Address offset: 0x550

y-intercept of 5th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x0000006A</b>	0	0	0	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	1	0
Id	RW	Field	Value Id	Value	Description																											
A	RW	B4			y-intercept of 5th piece wise linear function																											

27.1.15 B5

Address offset: 0x554

y-intercept of 6th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00003DD0</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	0	1	0	0	0
Id	RW	Field	Value Id	Value	Description																											
A	RW	B5			y-intercept of 6th piece wise linear function																											

27.1.16 T0

Address offset: 0x560

End point of 1st piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																										A	A	A	A	A	A	A	A
<b>Reset 0x000000E2</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	
Id	RW	Field	Value Id	Value	Description																												
A	RW	TO			End point of 1st piece wise linear function																												

27.1.17 T1

Address offset: 0x564

End point of 2nd piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Id																																A	A	A	A	A	A	A	A			
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																																					
A	RW	T1			End point of 2nd piece wise linear function																																					

27.1.18 T2

Address offset: 0x568

End point of 3rd piece wise linear function

27.1.19 T3

Address offset: 0x56C

End point of 4th piece wise linear function

27.1.20 T4

Address offset: 0x570

End point of 5th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
<b>Reset 0x00000050</b>	0 1 0 1 0 0 0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	T4			End point of 5th piece wise linear function

## 27.2 Electrical specification

### 27.2.1 Temperature Sensor Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TEMP}$	Time required for temperature measurement		36		$\mu\text{s}$
$T_{TEMP,RANGE}$	Temperature sensor range	-40		85	$^{\circ}\text{C}$
$T_{TEMP,ACC}$	Temperature sensor accuracy	-5		5	$^{\circ}\text{C}$
$T_{TEMP,RES}$	Temperature sensor resolution		0.25		$^{\circ}\text{C}$
$T_{TEMP,STB}$	Sample to sample stability at constant device temperature		+/-0.25		$^{\circ}\text{C}$
$T_{TEMP,OFFST}$	Sample offset at 25 $^{\circ}\text{C}$	-2.5		2.5	$^{\circ}\text{C}$

# 28 ECB — AES electronic codebook mode encryption

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

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

AES ECB features:

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

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

AES ECB can be stopped by triggering the STOPECB task.

## 28.1 Shared resources

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

## 28.2 EasyDMA

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

If the ECBDATAPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

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

## 28.3 ECB data structure

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

**Table 53: ECB data structure overview**

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

## 28.4 Registers

**Table 54: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4000E000	ECB	ECB	AES Electronic Code Book (ECB) mode block encryption	

**Table 55: Register Overview**

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

## **28.4.1 INTENSET**

Address offset: 0x304

## Enable interrupt

## 28.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

### 28.4.3 ECBDATAPTR

Address offset: 0x504

## ECB block encrypt memory pointers

## **28.5 Electrical specification**

### **28.5.1 ECB Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TEB}$	Run time per 16 byte block in all modes		6		$\mu\text{s}$

## 29 CCM — AES CCM mode encryption

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

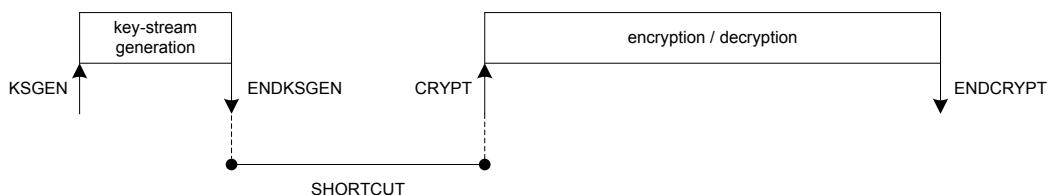
The CCM block generates an encrypted keystream that is applied to input data using the XOR operation and generates the 4 byte MIC field in one operation. The CCM and radio can be configured to work synchronously. The CCM will encrypt in time for transmission and decrypt after receiving bytes into memory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

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

The AES CCM supports three operations: key-stream generation, packet encryption, and packet decryption. All these operations are done in compliance with the *Bluetooth* specification.<sup>23</sup> A new key-stream must be generated before a new packet encryption or packet decryption operation can be started.

A key-stream is generated by triggering the KSGEN task. An ENDKSGEN event will be generated when the new key-stream has been generated. The key-stream will be stored in the AES CCM's temporary memory area, specified by the SCRATCHPTR, where it will be used in subsequent encryption and decryption operations.

Encryption is started by triggering the CRYPT task with the MODE register set to ENCRYPTION. Similarly, decryption is started by triggering the same task with MODE set to DECRYPTION. An ENDCRYPT event will be generated when packet encryption is completed as well as when packet decryption is completed, see [Figure 59: Key-stream generation followed by encryption or decryption. The shortcut is optional.](#) on page 266.



**Figure 59: Key-stream generation followed by encryption or decryption. The shortcut is optional.**

Key-stream generation, packet encryption, and packet decryption operations utilize the configuration specified in the data structure pointed to by the CNFPTTR pointer. It is necessary to configure this pointer and its underlying data structure, and the MODE register before the KSGEN task is triggered. It is also necessary to configure the INPTR pointer and the OUTPTR pointer before the CRYPT task is triggered.

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

The AES CCM supports different packet lengths, this is configured via the PACKETLENGTH field in the MODE register.

<sup>23</sup> Bluetooth AES CCM 128 bit block encryption, see *Bluetooth* Core specification Version 4.0.

## 29.1 Shared resources

The CCM shares registers and other resources with other peripherals that have the same ID as the CCM. The user must therefore disable all peripherals that have the same ID as the CCM before the CCM can be configured and used.

Disabling a peripheral that have the same ID as the CCM will not reset any of the registers that are shared with the CCM. It is therefore important to configure all relevant CCM registers explicitly to secure that it operates correctly.

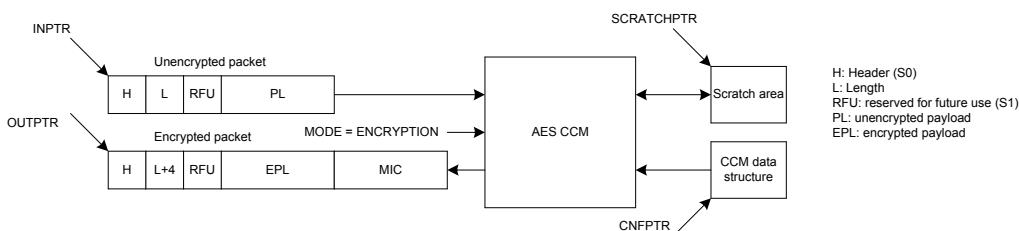
See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

## 29.2 Encryption

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

The AES CCM will also modify the length field of the packet to adjust for the appended MIC field, that is, add four bytes to the length, and store the resulting packet back into RAM at the address specified in the OUTPTR pointer, see [Figure 60: Encryption](#) on page 267.

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



**Figure 60: Encryption**

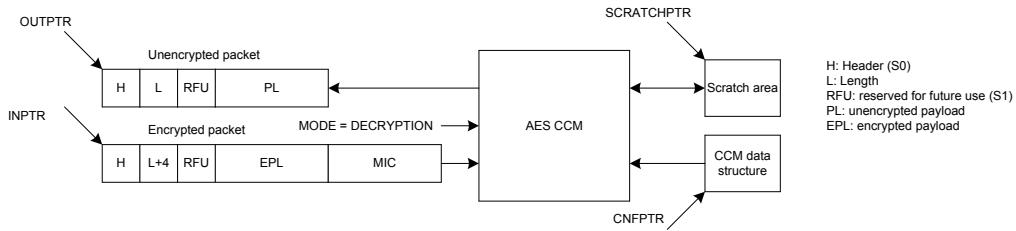
## 29.3 Decryption

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

The AES CCM will also modify the length field of the packet to adjust for the MIC field, that is, subtract four bytes from the length, and then store the decrypted packet into RAM at the address pointed to by the OUTPTR pointer, see [Figure 61: Decryption](#) on page 268.

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

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



**Figure 61: Decryption**

## 29.4 AES CCM and RADIO concurrent operation

The AES CCM is designed to run in parallel with the RADIO to enable on-the-fly encryption and decryption of RADIO packets without CPU involvement. To facilitate this, the RADIO has to be configured with specific settings.

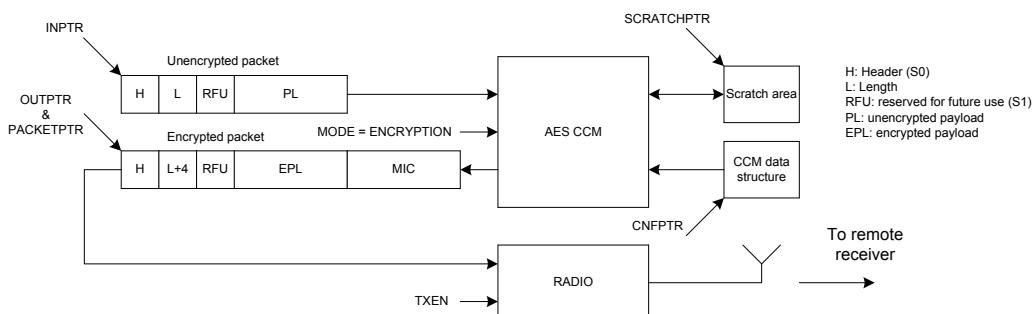
**Table 56: Radio configuration settings**

Radio parameter	Value	Description
PCNFO.SOLEN	1	Length of HEADER field in: <a href="#">Table 58: Data structure for unencrypted packet</a> on page 270 and <a href="#">Table 59: Data structure for encrypted packet</a> on page 270.
PCNFO.LFLEN	5 or 8	Length of LENGTH field in: <a href="#">Table 58: Data structure for unencrypted packet</a> on page 270 and <a href="#">Table 59: Data structure for encrypted packet</a> on page 270.
PCNFO.S1LEN	3 or 0	Length of the RFU field in: <a href="#">Table 58: Data structure for unencrypted packet</a> on page 270 and <a href="#">Table 59: Data structure for encrypted packet</a> on page 270. The combined length of LENGTH and RFU must always be 8 bit.
PCNFO.S1	Include	Always include the S1 field (RFU field) in RAM to secure that the same data structure can be used for PCNFO.S1LEN = 3 and PCNFO.S1LEN = 0: <a href="#">Table 58: Data structure for unencrypted packet</a> on page 270 and <a href="#">Table 59: Data structure for encrypted packet</a> on page 270.
MODE	Ble_1Mbit	Data rate. Must match CCM->MODE.DATARATE
PCNF1.BALEN	3	Length of address (32 bit)
CRCCNF.LEN	3	Length of CRC (24 bit)

## 29.5 Encrypting packets on-the-fly in radio transmit mode

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

The OUTPTR pointer in the AES CCM must therefore point to the same memory location as the PACKETPTR pointer in the RADIO, see [Figure 62: Configuration of on-the-fly encryption](#) on page 268.



**Figure 62: Configuration of on-the-fly encryption**

In order to match the RADIO's timing, the KSGEN task must be triggered no later than when the START task in the RADIO is triggered, in addition the shortcut between the ENDKSGEN event and the CRYPT task must be enabled. This use-case is illustrated in [Figure 63: On-the-fly encryption using a PPI connection](#) on page 269 using a PPI connection between the READY event in the RADIO and the KSGEN task in the AES CCM.

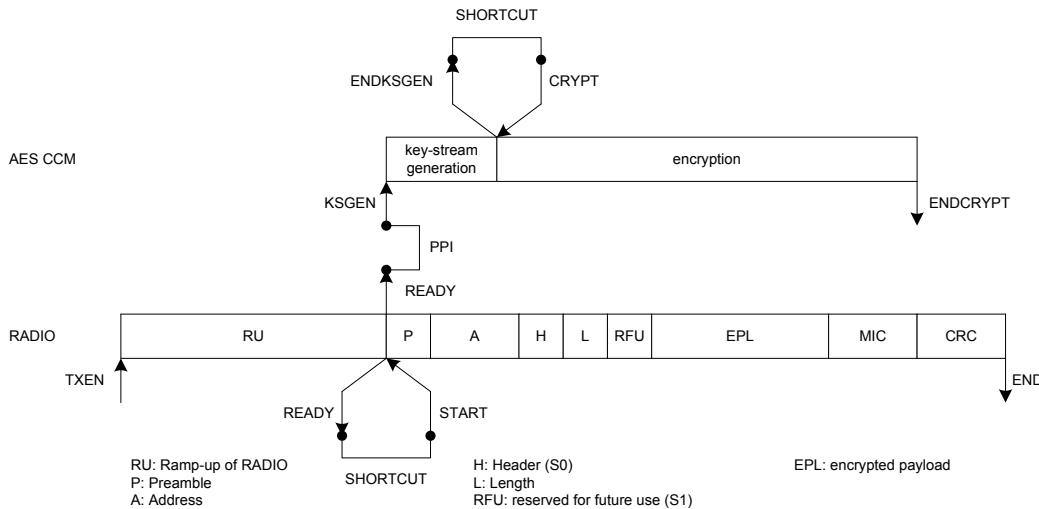


Figure 63: On-the-fly encryption using a PPI connection

## 29.6 Decrypting packets on-the-fly in radio receive mode

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

The INPTR pointer in the AES CCM must therefore point to the same memory location as the PACKETPTR pointer in the RADIO, see [Figure 64: Configuration of on-the-fly decryption](#) on page 269.

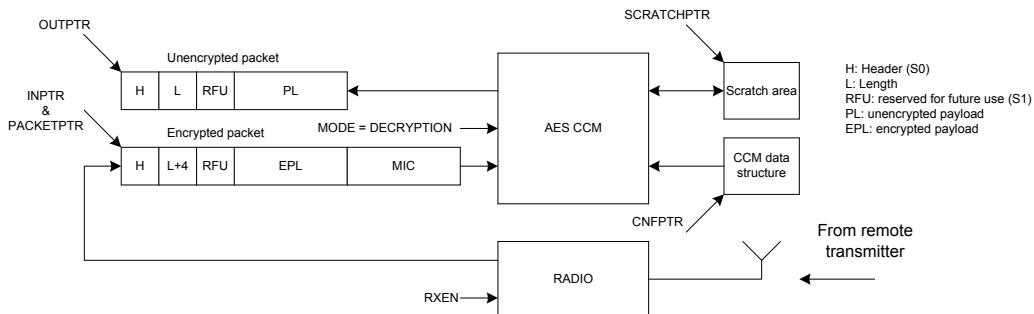
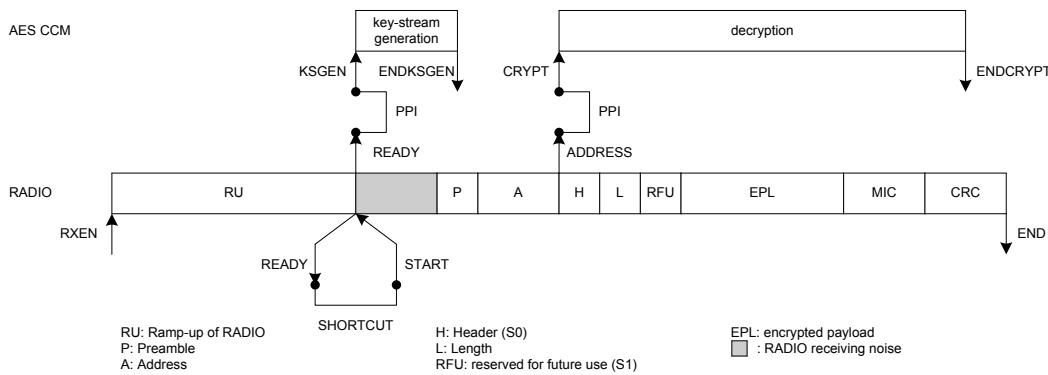


Figure 64: Configuration of on-the-fly decryption

In order to match the RADIO's timing, the KSGEN task must be triggered no later than when the START task in the RADIO is triggered. In addition, the CRYPT task must be triggered no earlier than when the ADDRESS event is generated by the RADIO.

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

This use-case is illustrated in [Figure 65: On-the-fly decryption using a PPI connection between the READY event in the RADIO and the KSGEN task in the AES CCM](#) on page 270 using a PPI connection between the ADDRESS event in the RADIO and the CRYPT task in the AES CCM. The KSGEN task is triggered from the READY event in the RADIO through a PPI connection.



**Figure 65: On-the-fly decryption using a PPI connection between the READY event in the RADIO and the KSGEN task in the AES CCM**

## 29.7 CCM data structure

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

**Table 57: CCM data structure overview**

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

The NONCE vector (as specified by the *Bluetooth Core Specification*) will be generated by hardware based on the information specified in the CCM data structure from [Table 57: CCM data structure overview](#) on page 270 .

**Table 58: Data structure for unencrypted packet**

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

**Table 59: Data structure for encrypted packet**

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

Property	Address offset	Description
<b>Important:</b> MIC is not added to empty packets		

## 29.8 EasyDMA and ERROR event

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

In some scenarios where the CPU and other DMA enabled peripherals are accessing the RAM at the same time, the CCM DMA could experience some bus conflicts which may also result in an error during encryption. If this happens, the ERROR event will be generated.

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

If the CNF PTR, SCRATCH PTR, INP TR and the OUTP TR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

## 29.9 Registers

**Table 60: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	CCM	CCM	AES CCM Mode Encryption	

**Table 61: Register Overview**

Register	Offset	Description
TASKS_KSGEN	0x000	Start generation of key-stream. This operation will stop by itself when completed.
TASKS_CRYPT	0x004	Start encryption/decryption. This operation will stop by itself when completed.
TASKS_STOP	0x008	Stop encryption/decryption
EVENTS_ENDKSGEN	0x100	Key-stream generation complete
EVENTS_ENDCRYPT	0x104	Encrypt/decrypt complete
EVENTS_ERROR	0x108	CCM error event
<b>SHORTS</b>	0x200	Shortcut register
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>MICSTATUS</b>	0x400	MIC check result
<b>ENABLE</b>	0x500	Enable
<b>MODE</b>	0x504	Operation mode
<b>CNF PTR</b>	0x508	Pointer to data structure holding AES key and NONCE vector
<b>INP TR</b>	0x50C	Input pointer
<b>OUTP TR</b>	0x510	Output pointer
<b>SCRATCH PTR</b>	0x514	Pointer to data area used for temporary storage

### 29.9.1 SHORTS

Address offset: 0x200

Shortcut register

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

## 29.9.2 INTENSET

Address offset: 0x304

## Enable interrupt

### 29.9.3 INTENCLR

Address offset: 0x308

Disable interrupt

## 29.9.4 MICSTATUS

Address offset: 0x400

## MIC check result

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
Reset	0x00000000																															
Id	RW	Field	Value Id																											Description		
A	R	MICSTATUS																											The result of the MIC check performed during the previous decryption operation			
		CheckFailed	0																										MIC check failed			
		CheckPassed	1																										MIC check passed			

## 29.9.5 ENABLE

Address offset: 0x500

Enable

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id	A	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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		Disabled				Enable or disable CCM																									
				0				Disable																									
				2				Enable																									

## 29.9.6 MODE

Address offset: 0x504

## Operation mode

## **29.9.7 CNF PTR**

Address offset: 0x508

Pointer to data structure holding AES key and NONCE vector

## **29.9.8 INPTR**

Address offset: 0x50C

## Input pointer

## 29.9.9 OUTPTR

Address offset: 0x510

## Output pointer

## 29.9.10 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

# 30 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*).

## 30.1 Shared resources

The AAR shares registers and other resources with the peripherals that have the same ID as the AAR. The user must therefore disable all peripherals that have the same ID as the AAR before the AAR can be configured and used.

Disabling a peripheral that have the same ID as the AAR will not reset any of the registers that are shared with the AAR. It is therefore important to configure all relevant AAR registers explicitly to secure that it operates correctly.

See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

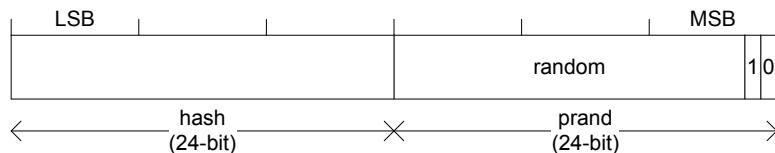
## 30.2 EasyDMA

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, ADDR PTR and the SCRATCHPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

## 30.3 Resolving a resolvable address

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



**Figure 66: Resolvable address**

To resolve an address the ADDR PTR 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*<sup>24</sup>. The time it takes to resolve an address may vary depending on where in the list the

<sup>24</sup> *Bluetooth Specification Version 4.0 [Vol 3]* chapter 10.8.2.3.

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.

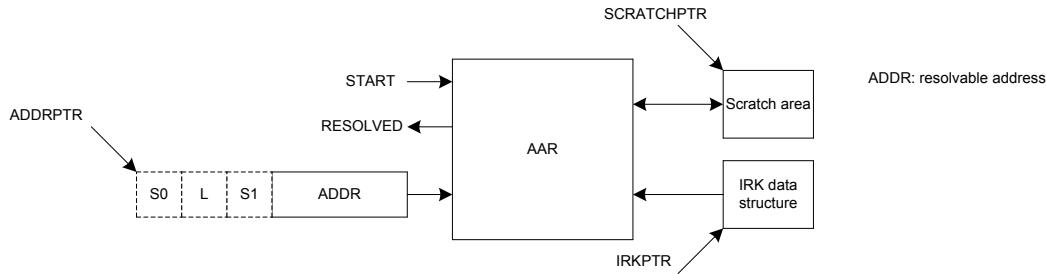


Figure 67: Address resolution with packet preloaded into RAM

### 30.4 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.

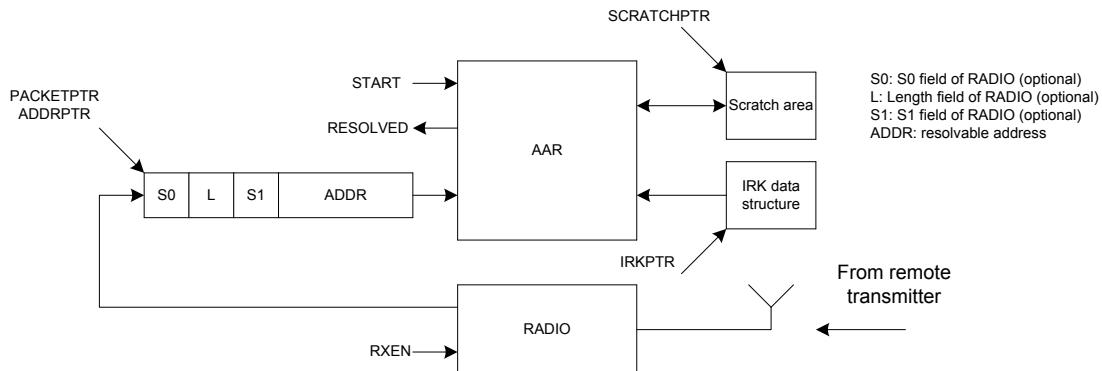


Figure 68: Address resolution with packet loaded into RAM by the RADIO

### 30.5 IRK data structure

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

Table 62: IRK data structure overview

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)

## 30.6 Registers

**Table 63: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	AAR	AAR	Acelerated Address Resolver	

**Table 64: Register Overview**

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

### 30.6.1 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																													
<b>Reset 0x00000000</b>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 <i>EVENTS_END</i>																											
				Set	1	Enable																											
				Disabled	0	Read: Disabled																											
				Enabled	1	Read: Enabled																											
B RW RESOLVED						Write '1' to Enable interrupt for RESOLVED event																											
						See <i>EVENTS_RESOLVED</i>																											
				Set	1	Enable																											
				Disabled	0	Read: Disabled																											
				Enabled	1	Read: Enabled																											
C RW NOTRESOLVED						Write '1' to Enable interrupt for NOTRESOLVED event																											
						See <i>EVENTS_NOTRESOLVED</i>																											
				Set	1	Enable																											
				Disabled	0	Read: Disabled																											
				Enabled	1	Read: Enabled																											

### 30.6.2 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		
Id	RW	Field	Value																								Description							
A RW END																											Write '1' to Disable interrupt for END event							
																											See <a href="#">EVENTS-END</a>							
																											Clear							
																											1							
																											Disable							
																											Disabled							
																											0							
																											Read: Disabled							
																											Enabled							
																											1							
B RW RESOLVED																											Write '1' to Disable interrupt for RESOLVED event							
																											See <a href="#">EVENTS-RESOLVED</a>							
																											Clear							
																											1							
																											Disable							
																											Disabled							
																											0							
																											Read: Enabled							
C RW NOTRESOLVED																											Write '1' to Disable interrupt for NOTRESOLVED event							
																											See <a href="#">EVENTS-NOTRESOLVED</a>							
																											Clear							
																											1							
																											Disable							
																											Disabled							
																											0							
																											Read: Disabled							
																											Enabled							
																											1							

### **30.6.3 STATUS**

Address offset: 0x400

## Resolution status

### **30.6.4 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																														
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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																											

## **30.6.5 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	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
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	NIRK	[1..16]		Number of Identity root keys available in the IRK data structure																											

### **30.6.6 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	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	0	0
Id	RW	Field	Value Id																								Description							
A	RW	IRKPTR																									Pointer to the IRK data structure							

### **30.6.7 ADDR PTR**

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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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)																											

### **30.6.8 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	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	0	
Id	RW	Field	Value Id																															
A	RW	SCRATCHPTR	Value																												Description			
A pointer to a scratch data area used for temporary storage during resolution. A space of minimum 3 bytes must be reserved.																														Pointer to a scratch data area used for temporary storage during resolution. A space of minimum 3 bytes must be reserved.				

## **30.7 Electrical specification**

### **30.7.1 AAR Electrical Specification**

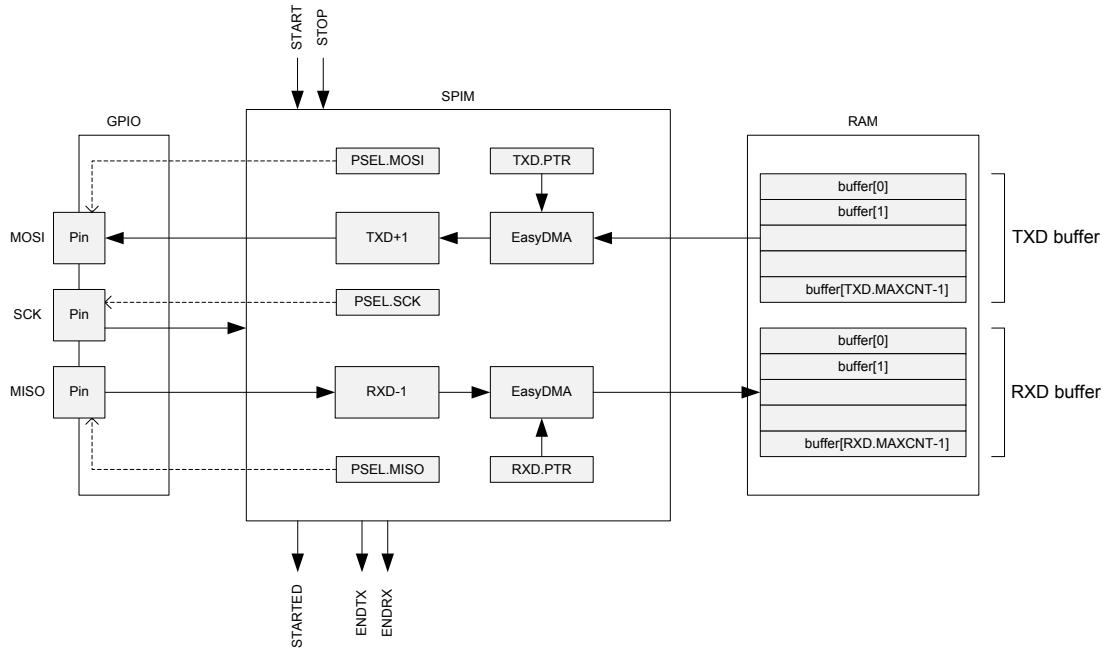
Symbol	Description	Min.	Typ.	Max.	Units
$t_{AAR,8}$	Time for address resolution of 8 IRKs		48		$\mu\text{s}$

# 31 SPIM — Serial peripheral interface master with EasyDMA

The SPI master can communicate with multiple slaves using individual chip select signals for each of the slave devices attached to a bus.

Listed here are the main features for the SPIM

- Three SPIM instances
- SPI mode 0-3
- EasyDMA direct transfer to/from RAM for both SPI Slave and SPI Master
- Individual selection of IO pin for each SPI signal



**Figure 69: SPIM — SPI master with EasyDMA**

The SPIM does not implement support for chip select directly. Therefore, the CPU must use available GPIOs to select the correct slave and control this independently of the SPI master. The SPIM supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

**Table 65: SPI modes**

Mode	Clock polarity CPOL	Clock phase CPHA
SPI_MODE_0 (Leading)	0 (Active High)	
SPI_MODE_0 (Leading)	1 (Active Low)	
SPI_MODE_1 (Trailing)	0 (Active High)	
SPI_MODE_1 (Trailing)	1 (Active Low)	

## 31.1 Shared resources

The SPI shares registers and other resources with other peripherals that have the same ID as the SPI. Therefore, the user must disable all peripherals that have the same ID as the SPI before the SPI can be configured and used.

Disabling a peripheral that has the same ID as the SPI will not reset any of the registers that are shared with the SPI. It is therefore important to configure all relevant SPI registers explicitly to secure that it operates correctly.

See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

## 31.2 EasyDMA

The SPI master implements EasyDMA for reading and writing of data packets from and to the DATA RAM without CPU involvement.

The RXD.PTR and TXD.PTR point to the RXD buffer (receive buffer) and TXD buffer (transmit buffer) respectively, see [Figure 69: SPIM — SPI master with EasyDMA](#) on page 280. RXD.MAXCNT and TXD.MAXCNT specify the maximum number of bytes allocated to the buffers.

The SPI master will automatically stop transmitting after TXD.MAXCNT bytes have been transmitted and RXD.MAXCNT bytes have been received. If TXD.MAXCNT is larger than RXD.MAXCNT, the superfluous received bytes will be ignored. If RXD.MAXCNT is larger than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register.

If the RXD.PTR and the TXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next transmission immediately after having received the STARTED event.

The ENDRX/ENDTX event indicate that EasyDMA has finished accessing respectively the RX/TX buffer in RAM. The END event gets generated when both RX and TX are finished accessing the buffers in RAM.

### 31.2.1 EasyDMA list

EasyDMA supports one list type.

The supported list type is:

- Array list

#### EasyDMA array list

The EasyDMA array list can be represented by the data structure `ArrayList_type`.

For illustration, see the code example below. This data structure includes only a buffer with size equal to `Channel.MAXCNT`. EasyDMA will use the `Channel.MAXCNT` register to determine when the buffer is full. Replace 'Channel' by the specific data channel you want to use, for instance '`NRF_SPIM->RXD`', '`NRF_SPIM->TXD`', '`NRF_TWIM->RXD`', etc.

The `Channel.MAXCNT` register cannot be specified larger than the actual size of the buffer. If `Channel.MAXCNT` is specified larger than the size of the buffer, the EasyDMA channel may overflow the buffer.

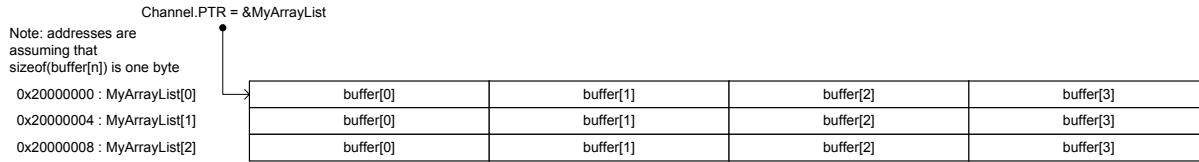
This array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

```
#define BUFFER_SIZE 4

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

ArrayList_type MyArrayList[3];
```

```
//replace 'Channel' below by the specific data channel you want to use,
//           for instance 'NRF_SPIM->RXD', 'NRF_TWIM->RXD', etc.
Channel.MAXCNT = BUFFER_SIZE;
Channel.PTR = &MyArrayList;
```



**Figure 70: EasyDMA array list**

### 31.3 SPI master transaction sequence

An SPI master transaction consists of a sequence started by the START task followed by a number of events, and finally the STOP task.

An SPI master transaction is started by triggering the START task. The ENDTX event will be generated when the transmitter has transmitted all bytes in the TXD buffer as specified in the TXD.MAXCNT register. The ENDRX event will be generated when the receiver has filled the RXD buffer, i.e. received the last possible byte as specified in the RXD.MAXCNT register.

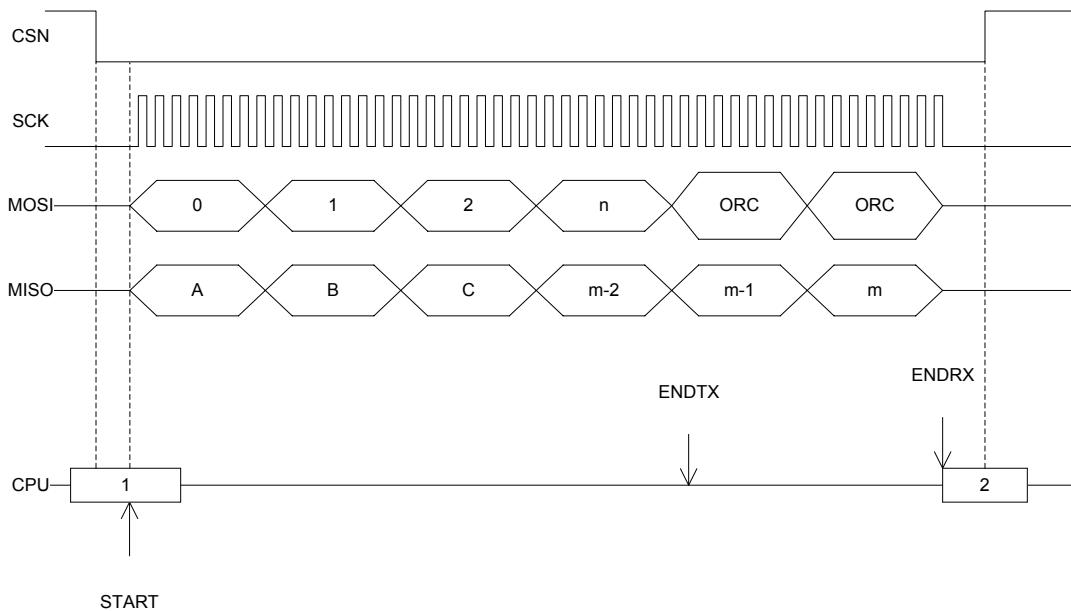
Following a START task, the SPI master will generate an END event when both ENDRX and ENDTX have been generated.

The SPI master is stopped by triggering the STOP task. A STOPPED event is generated when the SPI master has stopped.

If the ENDRX event has not already been generated when the SPI master has come to a stop, the SPI master will generate the ENDRX event explicitly even though the RX buffer is not full.

If the ENDTX event has not already been generated when the SPI master has come to a stop, the SPI master will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

The SPI master is a synchronous interface, and for every byte that is sent, a different byte will be received at the same time; this is illustrated in [Figure 71: SPI master transaction](#) on page 283.



**Figure 71: SPI master transaction**

### 31.4 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

### 31.5 Master mode pin configuration

The SCK, MOSI, and MISO signals associated with the SPI master are mapped to physical pins according to the configuration specified in the PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively.

The PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI master is enabled, and retained only as long as the device is in ON mode. PSEL.SCK, PSEL.MOSI and PSEL.MISO must only be configured when the SPI master is disabled.

To secure correct behavior in the SPI, the pins used by the SPI must be configured in the GPIO peripheral as described in [Table 66: GPIO configuration](#) on page 283 prior to enabling the SPI. This configuration must be retained in the GPIO for the selected IOs as long as the SPI is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 66: GPIO configuration**

SPI master signal	SPI master pin	Direction	Output value	Comments
SCK	As specified in PSEL.SCK	Output	Same as CONFIG.CPOL	
MOSI	As specified in PSEL.MOSI	Output	0	
MISO	As specified in PSEL.MISO	Input		Not applicable

## 31.6 Registers

**Table 67: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	SPIM	SPIMO	SPI master 0	
0x40004000	SPIM	SPIM1	SPI master 1	
0x40023000	SPIM	SPIM2	SPI master 2	

**Table 68: Register Overview**

Register	Offset	Description
TASKS_START	0x010	Start SPI transaction
TASKS_STOP	0x014	Stop SPI transaction
TASKS_SUSPEND	0x01C	Suspend SPI transaction
TASKS_RESUME	0x020	Resume SPI transaction
EVENTS_STOPPED	0x104	SPI transaction has stopped
EVENTS_ENDRX	0x110	End of RXD buffer reached
EVENTS_END	0x118	End of RXD buffer and TXD buffer reached
EVENTS_ENDTX	0x120	End of TXD buffer reached
EVENTS_STARTED	0x14C	Transaction started
<b>SHORTS</b>	0x200	Shortcut register
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable SPIM
PSEL.SCK	0x508	Pin select for SCK
PSEL.MOSI	0x50C	Pin select for MOSI signal
PSEL.MISO	0x510	Pin select for MISO signal
FREQUENCY	0x524	SPI frequency
RXD.PTR	0x534	Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last transaction
RXD.LIST	0x540	EasyDMA list type
TXD.PTR	0x544	Data pointer
TXD.MAXCNT	0x548	Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C	Number of bytes transferred in the last transaction
TXD.LIST	0x550	EasyDMA list type
CONFIG	0x554	Configuration register
ORC	0x5C0	Over-read character. Character clocked out in case of over-read of the TXD buffer.

### 31.6.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																													
Id	A																																																												
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																															
Id	RW	Field	Value Id																																																										
A	RW	END_START	Value																											Description																															
Shortcut between END event and START task																																																													
See <a href="#">EVENTS_END</a> and <a href="#">TASKS_START</a>																																																													
Disabled																																																													
Enabled																																																													

### 31.6.2 INTENSET

Address offset: 0x304

Enable interrupt

### 31.6.3 INTENCLR

Address offset: 0x308

## Disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E D C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
E RW STARTED					Write '1' to Disable interrupt for STARTED event
					See <a href="#">EVENTS_STARTED</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

### 31.6.4 ENABLE

Address offset: 0x500

Enable SPIM

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

### 31.6.5 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	B A A A A				
<b>Reset 0xFFFFFFFF</b>	1 1				
Id	RW	Field	Value Id	Value	Description
A RW PIN			[0..31]	Pin number	
B RW CONNECT					Connection
		Disconnected	1	Disconnect	
		Connected	0	Connect	

### 31.6.6 PSEL.MOSI

Address offset: 0x50C

Pin select for MOSI signal

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	B A A A A				
<b>Reset 0xFFFFFFFF</b>	1 1				
Id	RW	Field	Value Id	Value	Description
A RW PIN			[0..31]	Pin number	
B RW CONNECT					Connection
		Disconnected	1	Disconnect	
		Connected	0	Connect	

### 31.6.7 PSEL.MISO

Address offset: 0x510

Pin select for MISO signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **31.6.8 FREQUENCY**

Address offset: 0x524

## SPI frequency

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x04000000	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value																									Description					
A RW FREQUENCY			K125	0x02000000																								SPI master data rate						
			K250	0x04000000																								125 kbps						
			K500	0x08000000																								250 kbps						
			M1	0x10000000																								500 kbps						
			M2	0x20000000																								1 Mbps						
			M4	0x40000000																								2 Mbps						
			M8	0x80000000																								4 Mbps						
				0x100000000																								8 Mbps						

### **31.6.9 RXD.PTR**

Address offset: 0x534

## Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value																		Description										
A	RW	PTR																				Data pointer										

### **31.6.10 RXD.MAXCNT**

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	MAXCNT		Maximum number of bytes in receive buffer																												

### **31.6.11 RXD.AMOUNT**

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	AMOUNT			Number of bytes transferred in the last transaction																											

### **31.6.12 RXD.LIST**

Address offset: 0x540

## EasyDMA list type

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	LIST			List type																											
		Disabled	0	Disable EasyDMA list																												
		ArrayList	1	Use array list																												

### **31.6.13 TXD.PTR**

Address offset: 0x544

## Data pointer

### **31.6.14 TXD.MAXCNT**

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																										A	A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																												
A	RW	MAXCNT			Maximum number of bytes in transmit buffer																												

### **31.6.15 TXD.AMOUNT**

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	AMOUNT			Number of bytes transferred in the last transaction																											

### **31.6.16 TXD.LIST**

Address offset: 0x550

## EasyDMA list type

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	LIST			List type																											
		Disabled	0	Disable EasyDMA list																												
		ArrayList	1	Use array list																												

### 31.6.17 CONFIG

Address offset: 0x554

Configuration register

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id		C B A			
<b>Reset 0x00000000</b>		<b>0 0</b>			
Id	RW	Field	Value Id	Value	Description
A	RW	ORDER			Bit order
		MsbFirst	0		Most significant bit shifted out first
		LsbFirst	1		Least significant bit shifted out first
B	RW	CPHA			Serial clock (SCK) phase
		Leading	0		Sample on leading edge of clock, shift serial data on trailing edge
		Trailing	1		Sample on trailing edge of clock, shift serial data on leading edge
C	RW	CPOL			Serial clock (SCK) polarity
		ActiveHigh	0		Active high
		ActiveLow	1		Active low

### 31.6.18 ORC

Address offset: 0x5C0

Over-read character. Character clocked out in case and over-read of the TXD buffer.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id		A A A A A A A A			
<b>Reset 0x00000000</b>		<b>0 0</b>			
Id	RW	Field	Value Id	Value	Description
A	RW	ORC			Over-read character. Character clocked out in case and over-read of the TXD buffer.

## 31.7 Electrical specification

### 31.7.1 SPIM master interface

Symbol	Description	Min.	Typ.	Max.	Units
$f_{SPIM}$	Bit rates for SPIM <sup>25</sup>			$8^{26}$	Mbps
$I_{SPIM,2Mbps}$	Run current for SPIM, 2 Mbps		50		$\mu A$
$I_{SPIM,8Mbps}$	Run current for SPIM, 8 Mbps		50		$\mu A$
$I_{SPIM,IDLE}$	Idle current for SPIM (STARTed, no CSN activity)		1		$\mu A$
$t_{SPIM,START,LP}$	Time from START task to transmission started, low power mode			$t_{SPIM,START},$ +	$\mu s$
				$t_{START\_HFIN}$	
$t_{SPIM,START,CL}$	Time from START task to transmission started, constant latency mode	1			$\mu s$

### 31.7.2 Serial Peripheral Interface Master (SPIM) electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,CSC,8Mbps}$	SCK period at 8Mbps		125		ns
$t_{SPIM,CSC,4Mbps}$	SCK period at 4Mbps		250		ns
$t_{SPIM,CSC,2Mbps}$	SCK period at 2Mbps		500		ns

<sup>25</sup> Higher bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

<sup>26</sup> The actual maximum data rate depends on the slave's CLK to MISO and MOSI setup and hold timings.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,RCK,LD}$	SCK rise time, low drive <sup>a</sup>			$t_{RF,25pF}$	
$t_{SPIM,RCK,HD}$	SCK rise time, high drive <sup>a</sup>			$t_{HRF,25pF}$	
$t_{SPIM,FCK,LD}$	SCK fall time, low drive <sup>a</sup>			$t_{RF,25pF}$	
$t_{SPIM,FCK,HD}$	SCK fall time, high drive <sup>a</sup>			$t_{HRF,25pF}$	
$t_{SPIM,WHSCK}$	SCK high time <sup>a</sup>			$(0.5*t_{CSCK})$	
$t_{SPIM,WLSCK}$	SCK low time <sup>a</sup>			$(0.5*t_{CSCK})$	
$t_{SPIM,SUMI}$	MISO to CLK edge setup time	19		ns	
$t_{SPIM,HMI}$	CLK edge to MISO hold time	18		ns	
$t_{SPIM,VMO}$	CLK edge to MOSI valid			59	ns
$t_{SPIM,HMO}$	MOSI hold time after CLK edge	20		ns	

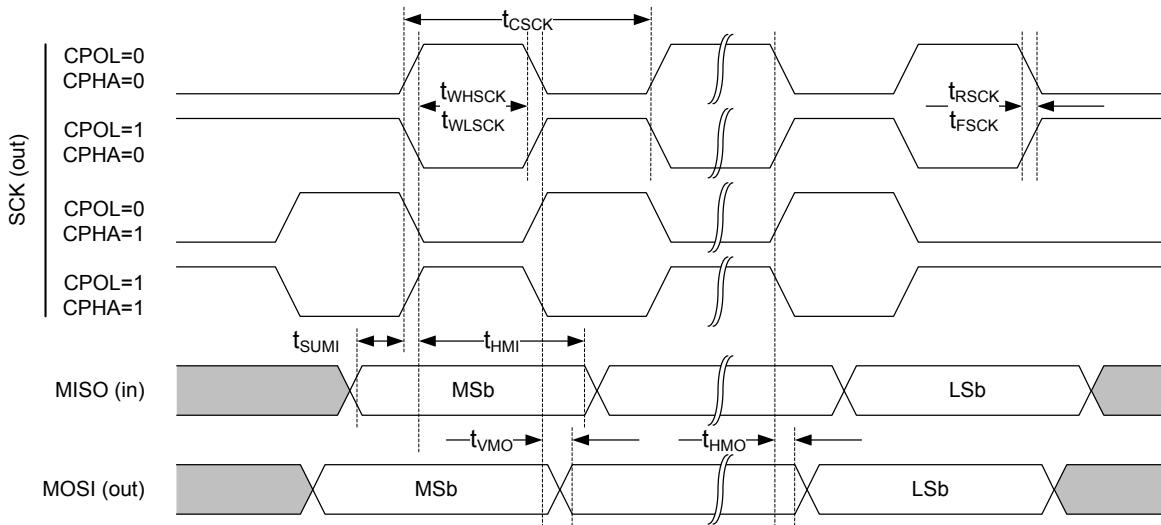
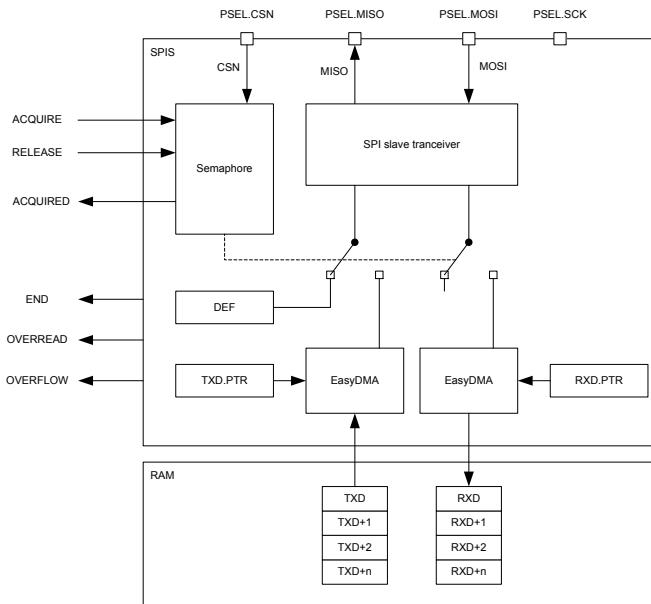


Figure 72: SPIM timing diagram

<sup>a</sup> At 25pF load, including GPIO pin capacitance, see GPIO spec.

## 32 SPIS — Serial peripheral interface slave with EasyDMA

SPI slave (SPIS) is implemented with EasyDMA support for ultra low power serial communication from an external SPI master. EasyDMA in conjunction with hardware-based semaphore mechanisms removes all real-time requirements associated with controlling the SPI slave from a low priority CPU execution context.



**Figure 73: SPI slave**

The SPIS supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

**Table 69: SPI modes**

Mode	Clock polarity CPOL	Clock phase CPHA
SPI_MODE0	0 (Leading)	0 (Active High)
SPI_MODE1	0 (Leading)	1 (Active Low)
SPI_MODE2	1 (Trailing)	0 (Active High)
SPI_MODE3	1 (Trailing)	1 (Active Low)

### 32.1 Shared resources

The SPI slave shares registers and other resources with other peripherals that have the same ID as the SPI slave. Therefore, you must disable all peripherals that have the same ID as the SPI slave before the SPI slave can be configured and used.

Disabling a peripheral that has the same ID as the SPI slave will not reset any of the registers that are shared with the SPI slave. It is important to configure all relevant SPI slave registers explicitly to secure that it operates correctly.

The Instantiation table in [Instantiation](#) on page 24 shows which peripherals have the same ID as the SPI slave.

### 32.2 EasyDMA

The SPI slave implements EasyDMA for reading and writing to and from the RAM. The END event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

### 32.3 SPI slave operation

SPI slave uses two memory pointers, RXD.PTR and TXD.PTR, that point to the RXD buffer (receive buffer) and TXD buffer (transmit buffer) respectively. Since these buffers are located in RAM, which can be accessed by both the SPI slave and the CPU, a hardware based semaphore mechanism is implemented to enable safe sharing.

See [Figure 74: SPI transaction when shortcut between END and ACQUIRE is enabled](#) on page 293.

Before the CPU can safely update the RXD.PTR and TXD.PTR pointers it must first acquire the SPI semaphore. The CPU can acquire the semaphore by triggering the ACQUIRE task and then receiving the ACQUIRED event. When the CPU has updated the RXD.PTR and TXD.PTR pointers the CPU must release the semaphore before the SPI slave will be able to acquire it. The CPU releases the semaphore by triggering the RELEASE task. This is illustrated in [Figure 74: SPI transaction when shortcut between END and ACQUIRE is enabled](#) on page 293. Triggering the RELEASE task when the semaphore is not granted to the CPU will have no effect.

The semaphore mechanism does not, at any time, prevent the CPU from performing read or write access to the RXD.PTR register, the TXD.PTR registers, or the RAM that these pointers are pointing to. The semaphore is only telling when these can be updated by the CPU so that safe sharing is achieved.

The semaphore is by default assigned to the CPU after the SPI slave is enabled. No ACQUIRED event will be generated for this initial semaphore handover. An ACQUIRED event will be generated immediately if the ACQUIRE task is triggered while the semaphore is assigned to the CPU.

The SPI slave will try to acquire the semaphore when CSN goes low. If the SPI slave does not manage to acquire the semaphore at this point, the transaction will be ignored. This means that all incoming data on MOSI will be discarded, and the DEF (default) character will be clocked out on the MISO line throughout the whole transaction. This will also be the case even if the semaphore is released by the CPU during the transaction. In case of a race condition where the CPU and the SPI slave try to acquire the semaphore at the same time, as illustrated in lifeline item 2 in [Figure 74: SPI transaction when shortcut between END and ACQUIRE is enabled](#) on page 293, the semaphore will be granted to the CPU.

If the SPI slave acquires the semaphore, the transaction will be granted. The incoming data on MOSI will be stored in the RXD buffer and the data in the TXD buffer will be clocked out on MISO.

When a granted transaction is completed and CSN goes high, the SPI slave will automatically release the semaphore and generate the END event.

As long as the semaphore is available the SPI slave can be granted multiple transactions one after the other. If the CPU is not able to reconfigure the TXD.PTR and RXD.PTR between granted transactions, the same TX data will be clocked out and the RX buffers will be overwritten. To prevent this from happening, the END\_ACQUIRE shortcut can be used. With this shortcut enabled the semaphore will be handed over to the CPU automatically after the granted transaction has completed, giving the CPU the ability to update the TXPTR and RXPTR between every granted transaction.

If the CPU tries to acquire the semaphore while it is assigned to the SPI slave, an immediate handover will not be granted. However, the semaphore will be handed over to the CPU as soon as the SPI slave has released the semaphore after the granted transaction is completed. If the END\_ACQUIRE shortcut is enabled and the CPU has triggered the ACQUIRE task during a granted transaction, only one ACQUIRE request will be served following the END event.

The MAXRX register specifies the maximum number of bytes the SPI slave can receive in one granted transaction. If the SPI slave receives more than MAXRX number of bytes, an OVERFLOW will be indicated in the STATUS register and the incoming bytes will be discarded.

The MAXTX parameter specifies the maximum number of bytes the SPI slave can transmit in one granted transaction. If the SPI slave is forced to transmit more than MAXTX number of bytes, an OVERREAD will be indicated in the STATUS register and the ORC character will be clocked out.

The RXD.AMOUNT and TXD.AMOUNT registers are updated when a granted transaction is completed. The TXD.AMOUNT register indicates how many bytes were read from the TX buffer in the last transaction, that is, ORC (over-read) characters are not included in this number. Similarly, the RXD.AMOUNT register indicates how many bytes were written into the RX buffer in the last transaction.

The ENDRX event is generated when the RX buffer has been filled.

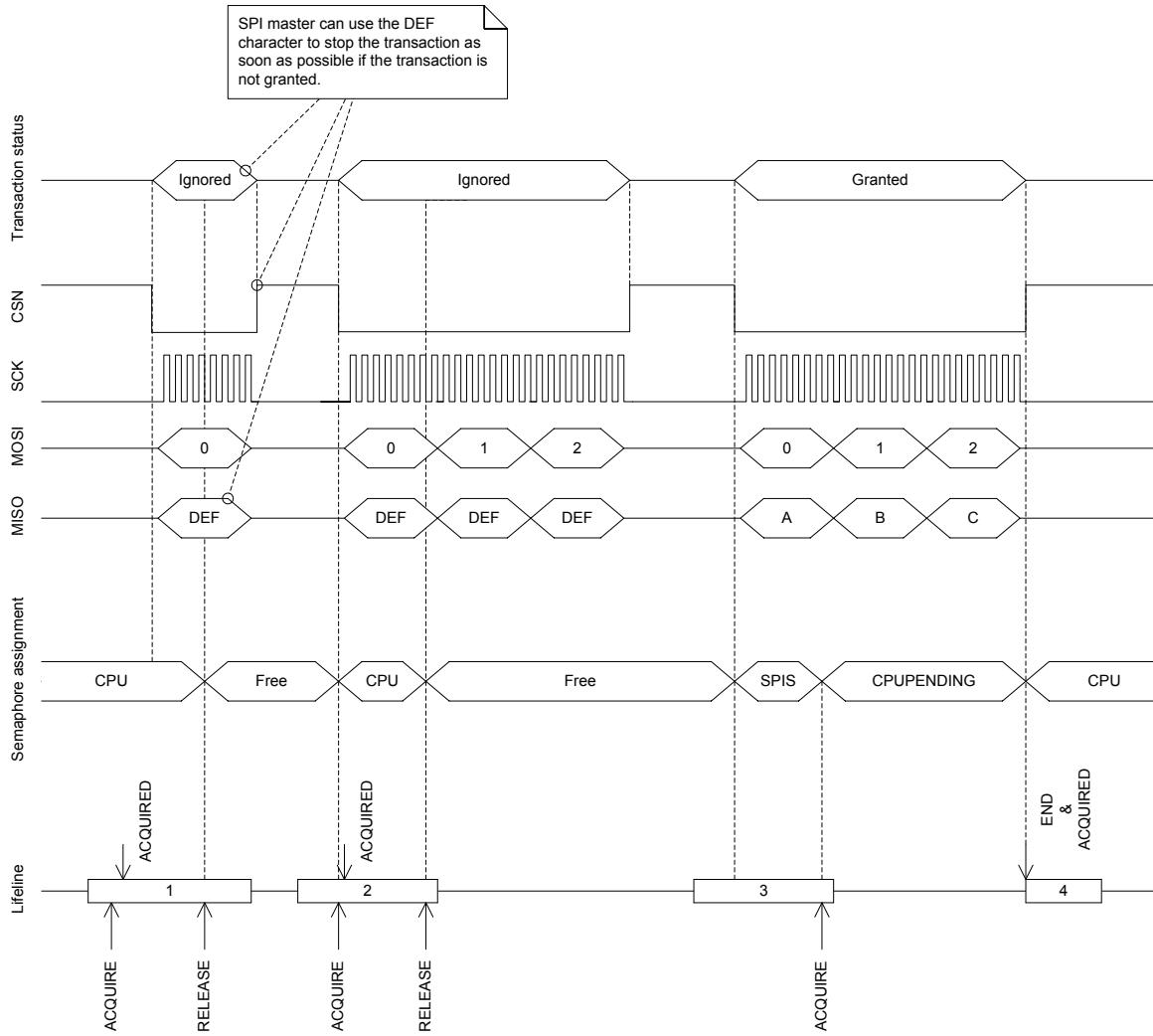


Figure 74: SPI transaction when shortcut between END and ACQUIRE is enabled

## 32.4 Pin configuration

The CSN, SCK, MOSI, and MISO signals associated with the SPI slave are mapped to physical pins according to the configuration specified in the PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively. If the CONNECT field of any of these registers is set to Disconnected, the associated SPI slave signal will not be connected to any physical pins.

The PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI slave is enabled, and retained only as long as the device is in System ON mode, see [POWER — Power supply](#) on page 78 chapter for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN\_CNF[n] register. PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI slave is disabled.

To secure correct behavior in the SPI slave, the pins used by the SPI slave must be configured in the GPIO peripheral as described in [Table 70: GPIO configuration before enabling peripheral](#) on page 294 before enabling the SPI slave. This is to secure that the pins used by the SPI slave are driven correctly if the SPI

slave itself is temporarily disabled, or if the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the SPI slave is to be recognized by an external SPI master.

The MISO line is set in high impedance as long as the SPI slave is not selected with CSN.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 70: GPIO configuration before enabling peripheral**

SPI signal	SPI pin	Direction	Output value	Comment
CSN	As specified in PSEL.CSN	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
MOSI	As specified in PSEL.MOSI	Input	Not applicable	
MISO	As specified in PSEL.MISO	Input	Not applicable	Emulates that the SPI slave is not selected.

## 32.5 Registers

**Table 71: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	SPIS	SPISO	SPI slave 0	
0x40004000	SPIS	SPIS1	SPI slave 1	
0x40023000	SPIS	SPIS2	SPI slave 2	

**Table 72: Register Overview**

Register	Offset	Description	
TASKS_ACQUIRE	0x024	Acquire SPI semaphore	
TASKS_RELEASE	0x028	Release SPI semaphore, enabling the SPI slave to acquire it	
EVENTS_END	0x104	Granted transaction completed	
EVENTS_ENDRX	0x110	End of RXD buffer reached	
EVENTS_ACQUIRED	0x128	Semaphore acquired	
<i>SHORTS</i>	0x200	Shortcut register	
<i>INTENSET</i>	0x304	Enable interrupt	
<i>INTENCLR</i>	0x308	Disable interrupt	
<i>SEMSTAT</i>	0x400	Semaphore status register	
<i>STATUS</i>	0x440	Status from last transaction	
<i>ENABLE</i>	0x500	Enable SPI slave	
<i>PSEL.SCK</i>	0x508	Pin select for SCK	Deprecated
<i>PSEL.MISO</i>	0x50C	Pin select for MISO	Deprecated
<i>PSEL.MOSI</i>	0x510	Pin select for MOSI	Deprecated
<i>PSEL.CSN</i>	0x514	Pin select for CSN	Deprecated
<i>PSEL.SCK</i>	0x508	Pin select for SCK	
<i>PSEL.MISO</i>	0x50C	Pin select for MISO signal	
<i>PSEL.MOSI</i>	0x510	Pin select for MOSI signal	
<i>PSEL.CSN</i>	0x514	Pin select for CSN signal	
<i>RXD.PTR</i>	0x534	RXD data pointer	Deprecated
<i>MAXRX</i>	0x538	Maximum number of bytes in receive buffer	Deprecated
<i>AMOUNTRX</i>	0x53C	Number of bytes received in last granted transaction	Deprecated
<i>RXD.PTR</i>	0x534	RXD data pointer	
<i>RXD.MAXCNT</i>	0x538	Maximum number of bytes in receive buffer	
<i>RXD.AMOUNT</i>	0x53C	Number of bytes received in last granted transaction	
<i>TXD.PTR</i>	0x544	TXD data pointer	Deprecated
<i>MAXTX</i>	0x548	Maximum number of bytes in transmit buffer	Deprecated
<i>AMOUNTTX</i>	0x54C	Number of bytes transmitted in last granted transaction	Deprecated
<i>TXD.PTR</i>	0x544	TXD data pointer	
<i>TXD.MAXCNT</i>	0x548	Maximum number of bytes in transmit buffer	
<i>TXD.AMOUNT</i>	0x54C	Number of bytes transmitted in last granted transaction	
<i>CONFIG</i>	0x554	Configuration register	

Register	Offset	Description
<i>DEF</i>	0x55C	Default character. Character clocked out in case of an ignored transaction.
<i>ORC</i>	0x5C0	Over-read character

### 32.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id	A			
Reset 0x00000000	0 0			
Id	RW	Field	Value	Description
A	RW	END_ACQUIRE		Shortcut between END event and ACQUIRE task
				See <a href="#">EVENTS_END</a> and <a href="#">TASKS_ACQUIRE</a>
		Disabled	0	Disable shortcut
		Enabled	1	Enable shortcut

### 32.5.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id	C B A			
Reset 0x00000000	0 0			
Id	RW	Field	Value	Description
A	RW	END		Write '1' to Enable interrupt for END event
				See <a href="#">EVENTS_END</a>
		Set	1	Enable
		Disabled	0	Read: Disabled
		Enabled	1	Read: Enabled
B	RW	ENDRX		Write '1' to Enable interrupt for ENDRX event
				See <a href="#">EVENTS_ENDRX</a>
		Set	1	Enable
		Disabled	0	Read: Disabled
		Enabled	1	Read: Enabled
C	RW	ACQUIRED		Write '1' to Enable interrupt for ACQUIRED event
				See <a href="#">EVENTS_ACQUIRED</a>
		Set	1	Enable
		Disabled	0	Read: Disabled
		Enabled	1	Read: Enabled

### 32.5.3 INTENCLR

Address offset: 0x308

Disable interrupt

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

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_ENDRX</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
C RW ACQUIRED					Write '1' to Disable interrupt for ACQUIRED event
					See <a href="#">EVENTS_ACQUIRED</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

### 32.5.4 SEMSTAT

Address offset: 0x400

Semaphore status register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A				
<b>Reset 0x00000001</b>	0 1				
Id	RW	Field	Value Id	Value	Description
A R SEMSTAT					Semaphore status
		Free	0	Semaphore is free	
		CPU	1	Semaphore is assigned to CPU	
		SPIS	2	Semaphore is assigned to SPI slave	
		CPUPending	3	Semaphore is assigned to SPI but a handover to the CPU is pending	

### 32.5.5 STATUS

Address offset: 0x440

Status from last transaction

Individual bits are cleared by writing a '1' to the bits that shall be cleared

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A RW OVERREAD					TX buffer over-read detected, and prevented
		NotPresent	0	Read: error not present	
		Present	1	Read: error present	
		Clear	1	Write: clear error on writing '1'	
B RW OVERFLOW					RX buffer overflow detected, and prevented
		NotPresent	0	Read: error not present	
		Present	1	Read: error present	
		Clear	1	Write: clear error on writing '1'	

### 32.5.6 ENABLE

Address offset: 0x500

Enable SPI slave

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A RW ENABLE					Enable or disable SPI slave

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
			Disabled	0	Disable SPI slave																											
			Enabled	2	Enable SPI slave																											

### **32.5.7 PSELSCK ( Deprecated )**

Address offset: 0x508

## Pin select for SCK

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	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	PSELCK		[0..31]	Pin number configuration for SPI SCK signal																											
			Disconnected	0xFFFFFFFF	Disconnect																											

### **32.5.8 PSELMISO ( Deprecated )**

Address offset: 0x50C

## Pin select for MISO

### **32.5.9 PSELMOSSI ( Deprecated )**

Address offset: 0x510

## Pin select for MOSI

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	1	1
Id	RW	Field	Value																										Description					
A	RW	PSELMO <sup>I</sup>	[0..31]																										Pin number configuration for SPI MOSI signal					
		Disconnected	0xFFFFFFFF																										Disconnect					

### **32.5.10 PSELCSN ( Deprecated )**

Address offset: 0x514

Pin select for CSN

### **32.5.11 PSEL.SCK**

Address offset: 0x508

## Pin select for SCK

### **32.5.12 PSEL.MISO**

Address offset: 0x50C

Pin select for MISO signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																										A	A	A	A	A	
Reset 0xFFFFFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value		Description																										
A	RW	PIN		[0..31]		Pin number																										
B	RW	CONNECT				Connection																										
			Disconnected	1		Disconnect																										
			Connected	0		Connect																										

### 32.5.13 PSEL.MOSI

Address offset: 0x510

## Pin select for MOSI signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **32.5.14 PSEL.CSN**

Address offset: 0x514

### Pin select for CSN signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																										A	A	A	A	A	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **32.5.15 RXDPTR ( Deprecated )**

Address offset: 0x534

## RXD data pointer

### **32.5.16 MAXRX ( Deprecated )**

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	MAXRX			Maximum number of bytes in receive buffer																											

### **32.5.17 AMOUNTRX ( Deprecated )**

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value																Description												
A	R	AMOUNTRX																		Number of bytes received in the last granted transaction												

### **32.5.18 RXD.PTR**

Address offset: 0x534

## RXD data pointer

### 32.5.19 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

### 32.5.20 RXD-AMOUNT

Address offset: 0x53C

Number of bytes received in last granted transaction

### **32.5.21 TXDPTR ( Deprecated )**

Address offset: 0x544

TXD data pointer

### **32.5.22 MAXTX ( Deprecated )**

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	MAXTX			Maximum number of bytes in transmit buffer																											

### **32.5.23 AMOUNTTX ( Deprecated )**

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

### **32.5.24 TXD.PTR**

Address offset: 0x544

TXD data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PTR			TXD data pointer																											

### 32.5.25 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																										A	A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																											
A	RW	MAXCNT				Maximum number of bytes in transmit buffer																											

### **32.5.26 TXD.AMOUNT**

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	R	AMOUNT			Number of bytes transmitted in last granted transaction

### 32.5.27 CONFIG

Address offset: 0x554

Configuration register

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	C B A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	ORDER			Bit order
		MsbFirst	0		Most significant bit shifted out first
		LsbFirst	1		Least significant bit shifted out first
B	RW	CPHA			Serial clock (SCK) phase
		Leading	0		Sample on leading edge of clock, shift serial data on trailing edge
		Trailing	1		Sample on trailing edge of clock, shift serial data on leading edge
C	RW	CPOL			Serial clock (SCK) polarity
		ActiveHigh	0		Active high
		ActiveLow	1		Active low

### 32.5.28 DEF

Address offset: 0x55C

Default character. Character clocked out in case of an ignored transaction.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DEF			Default character. Character clocked out in case of an ignored transaction.

### 32.5.29 ORC

Address offset: 0x5C0

Over-read character

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	ORC			Over-read character. Character clocked out after an over-read of the transmit buffer.

## 32.6 Electrical specification

### 32.6.1 SPIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{SPIS}$	Bit rates for SPIS <sup>27</sup>			$8^{28}$	Mbps
$I_{SPIS,2Mbps}$	Run current for SPIS, 2 Mbps		45		$\mu A$
$I_{SPIS,8Mbps}$	Run current for SPIS, 8 Mbps		45		$\mu A$
$I_{SPIS,IDLE}$	Idle current for SPIS (STARTed, no CSN activity)		1		$\mu A$
$t_{SPIS,LP,START}$	Time from RELEASE task to ready to receive/transmit (CSN active), Low power mode		$t_{SPIS,CL,STAR}$		$\mu s$
			+		
				$t_{START\_HFIN}$	
$t_{SPIS,CL,START}$	Time from RELEASE task to receive/transmit (CSN active), Constant latency mode	0.125			$\mu s$

### 32.6.2 Serial Peripheral Interface Slave (SPIS) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,CSCKIN,8Mbps}$	SCK input period at 8Mbps		125		ns
$t_{SPIS,CSCKIN,4Mbps}$	SCK input period at 4Mbps		250		ns
$t_{SPIS,CSCKIN,2Mbps}$	SCK input period at 2Mbps		500		ns
$t_{SPIS,RFSCKIN}$	SCK input rise/fall time		30		ns
$t_{SPIS,WHSCKIN}$	SCK input high time	30			ns
$t_{SPIS,WLSCKIN}$	SCK input low time	30			ns
$t_{SPIS,SUCSN,LP}$	CSN to CLK setup time, Low power mode		$t_{SPIS,SUCSN,LP}$		ns
			+		
			$t_{START\_HFIN}$		
$t_{SPIS,SUCSN,CL}$	CSN to CLK setup time, Constant latency mode	1000			ns
$t_{SPIS,HCSN}$	CLK to CSN hold time	2000			ns
$t_{SPIS,ASO}$	CSN to MISO driven <sup>a</sup>		1000		ns
$t_{SPIS,DISO}$	CSN to MISO disabled <sup>a</sup>		68		ns
$t_{SPIS,CWH}$	CSN inactive time	300			ns
$t_{SPIS,VSO}$	CLK edge to MISO valid		19		ns
$t_{SPIS,HSO}$	MISO hold time after CLK edge	18 <sup>29</sup>			ns
$t_{SPIS,SUSI}$	MOSI to CLK edge setup time	59			ns
$t_{SPIS,HSI}$	CLK edge to MOSI hold time	20			ns

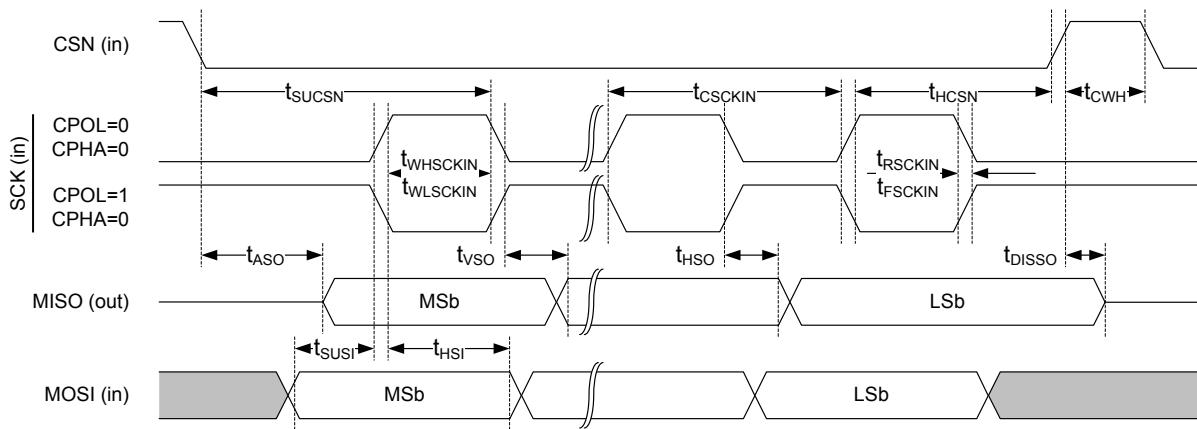


Figure 75: SPIS timing diagram

<sup>27</sup> Higher bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

<sup>28</sup> The actual maximum data rate depends on the master's CLK to MISO and MOSI setup and hold timings.

<sup>a</sup> At 25pF load, including GPIO capacitance, see GPIO spec.

<sup>29</sup> This is to ensure compatibility to SPI masters sampling MISO on the same edge as MOSI is output

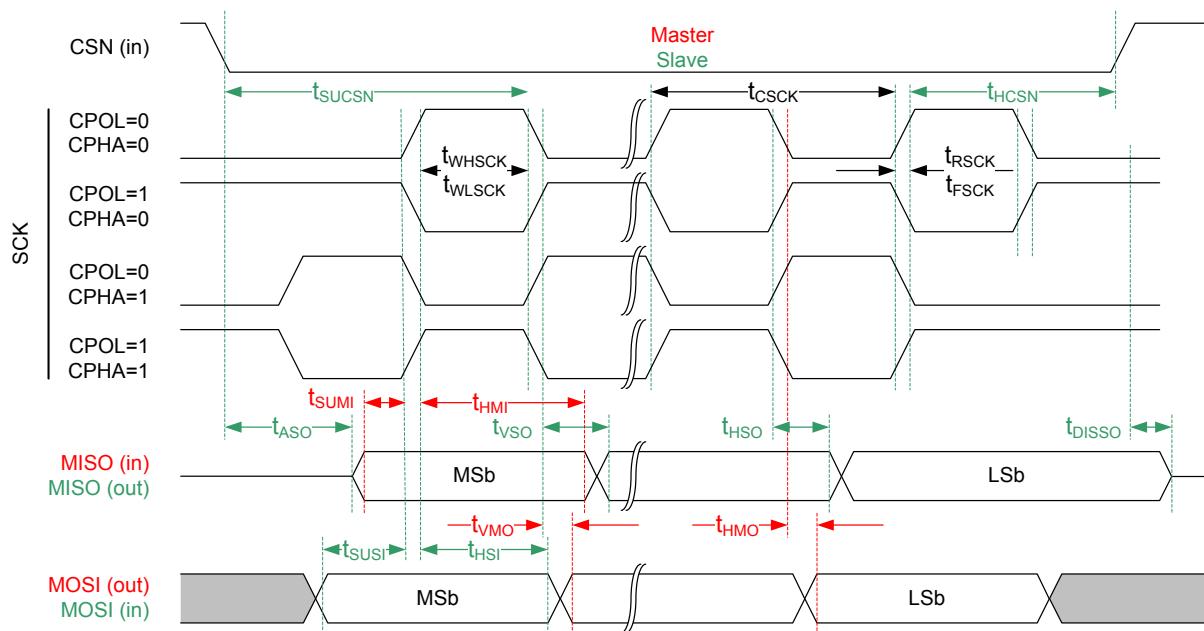


Figure 76: Common SPIM and SPIS timing diagram

# 33 TWIM — I<sup>2</sup>C compatible two-wire interface master with EasyDMA

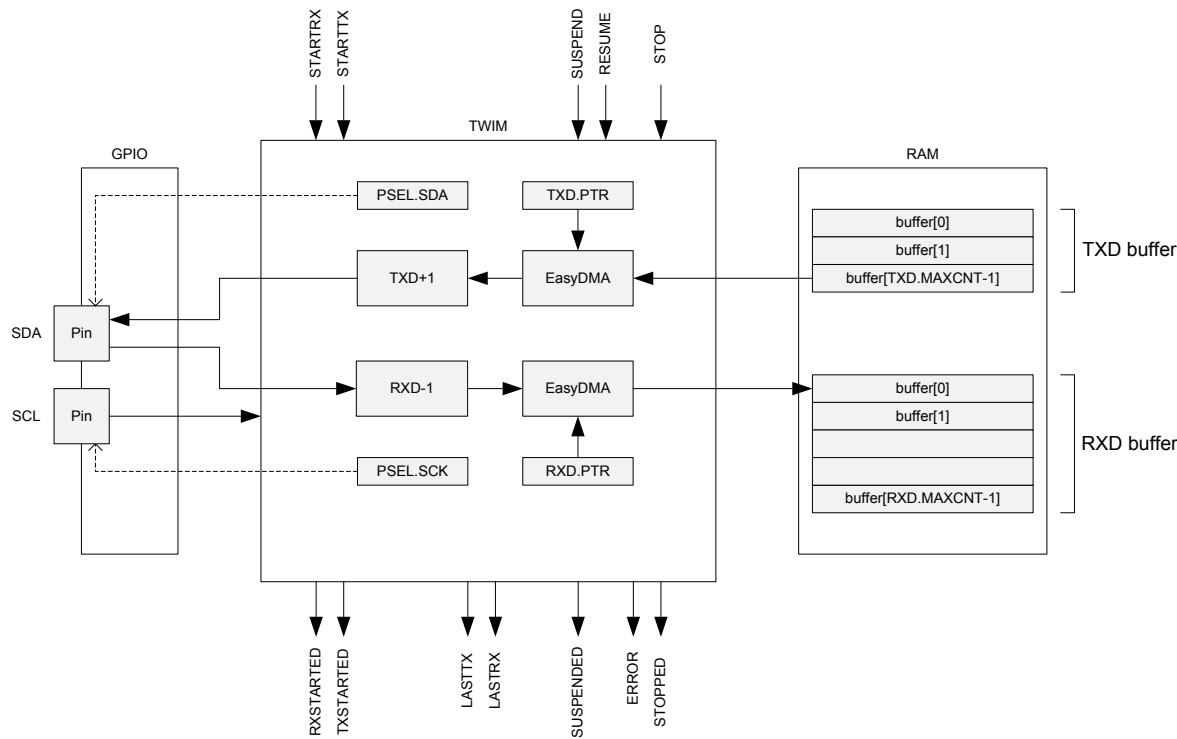
TWI master with EasyDMA (TWIM) is a two-wire half-duplex master which can communicate with multiple slave devices connected to the same bus

Listed here are the main features for TWIM:

- I<sup>2</sup>C compatible
- 100 kbps, 250 kbps, or 400 kbps
- Support for clock stretching
- EasyDMA

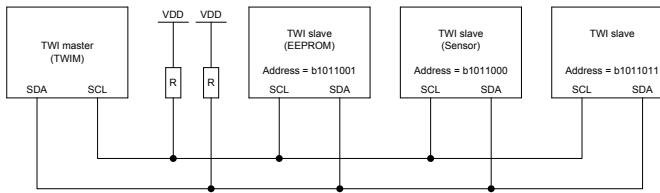
The two-wire interface can communicate with a bi-directional wired-AND bus with two lines (SCL, SDA). The protocol makes it possible to interconnect up to 127 individually addressable devices. TWIM is not compatible with CBUS.

The GPIOs used for each two-wire interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.



**Figure 77: TWI master with EasyDMA**

A typical TWI setup consists of one master and one or more slaves. For an example, see [Figure 78: A typical TWI setup comprising one master and three slaves](#) on page 305. This TWIM is only able to operate as a single master on the TWI bus. Multi-master bus configuration is not supported.



**Figure 78: A typical TWI setup comprising one master and three slaves**

This TWI master supports clock stretching performed by the slaves. The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task. The TWI master will generate a STOPPED event when it has stopped following a STOP task.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

After the TWI master is started, the STARTTX task or the STARTRX task should not be triggered again before the TWI master has stopped, i.e. following a LASTRX, LASTTX or STOPPED event.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

### 33.1 Shared resources

The TWI master shares registers and other resources with other peripherals that have the same ID as the TWI master. Therefore, you must disable all peripherals that have the same ID as the TWI master before the TWI master can be configured and used.

Disabling a peripheral that has the same ID as the TWI master will not reset any of the registers that are shared with the TWI master. It is therefore important to configure all relevant registers explicitly to secure that the TWI master operates correctly.

The Instantiation table in [Instantiation](#) on page 24 shows which peripherals have the same ID as the TWI.

### 33.2 EasyDMA

The TWI master implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

#### 33.2.1 EasyDMA list

EasyDMA supports one list type.

The supported list type is:

- Array list

##### EasyDMA array list

The EasyDMA array list can be represented by the data structure `ArrayList_type`.

For illustration, see the code example below. This data structure includes only a buffer with size equal to `Channel.MAXCNT`. EasyDMA will use the `Channel.MAXCNT` register to determine when the buffer is full. Replace 'Channel' by the specific data channel you want to use, for instance '`NRF_SPIM->RXD`', '`NRF_SPIM->TXD`', '`NRF_TWIM->RXD`', etc.

The Channel.MAXCNT register cannot be specified larger than the actual size of the buffer. If Channel.MAXCNT is specified larger than the size of the buffer, the EasyDMA channel may overflow the buffer.

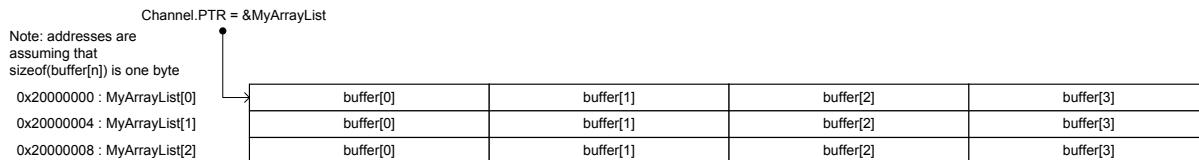
This array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

```
#define BUFFER_SIZE 4

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

ArrayList_type MyArrayList[3];

//replace 'Channel' below by the specific data channel you want to use,
//for instance 'NRF_SPIM->RXD', 'NRF_TWIM->RXD', etc.
Channel.MAXCNT = BUFFER_SIZE;
Channel.PTR = &MyArrayList;
```



**Figure 79: EasyDMA array list**

### 33.3 Master write sequence

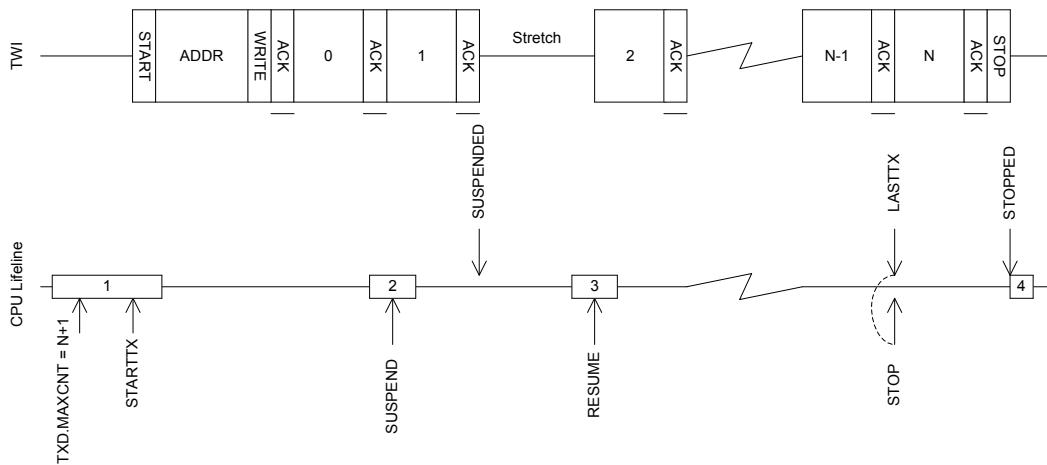
A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes found in the transmit buffer located in RAM at the address specified in the TXD.PTR register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave.

A typical TWI master write sequence is illustrated in [Figure 80: TWI master writing data to a slave](#) on page 307. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.



**Figure 80: TWI master writing data to a slave**

The TWI master will generate a LASTTX event when it starts to transmit the last byte, this is illustrated in [Figure 80: TWI master writing data to a slave](#) on page 307

The TWI master is stopped by triggering the STOP task, this task should be triggered during the transmission of the last byte to secure that the TWI will stop as fast as possible after sending the last byte. It is safe to use the shortcut between LASTTX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the whole RAM buffer has been sent, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

### 33.4 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1). The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

Data received will be stored in RAM at the address specified in the RXD.PTR register. The TWI master will generate an ACK after all but the last byte received from the slave. The TWI master will generate a NACK after the last byte received to indicate that the read sequence shall stop.

A typical TWI master read sequence is illustrated in [Figure 81: The TWI master reading data from a slave](#) on page 308. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

The TWI master will generate a LASTRX event when it is ready to receive the last byte, this is illustrated in [Figure 81: The TWI master reading data from a slave](#) on page 308. If RXD.MAXCNT > 1 the LASTRX event is generated after sending the ACK of the previously received byte. If RXD.MAXCNT = 1 the LASTRX event is generated after receiving the ACK following the address and READ bit.

The TWI master is stopped by triggering the STOP task, this task must be triggered before the NACK bit is supposed to be transmitted. The STOP task can be triggered at any time during the reception of the last byte. It is safe to use the shortcut between LASTRX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the RAM buffer is full, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

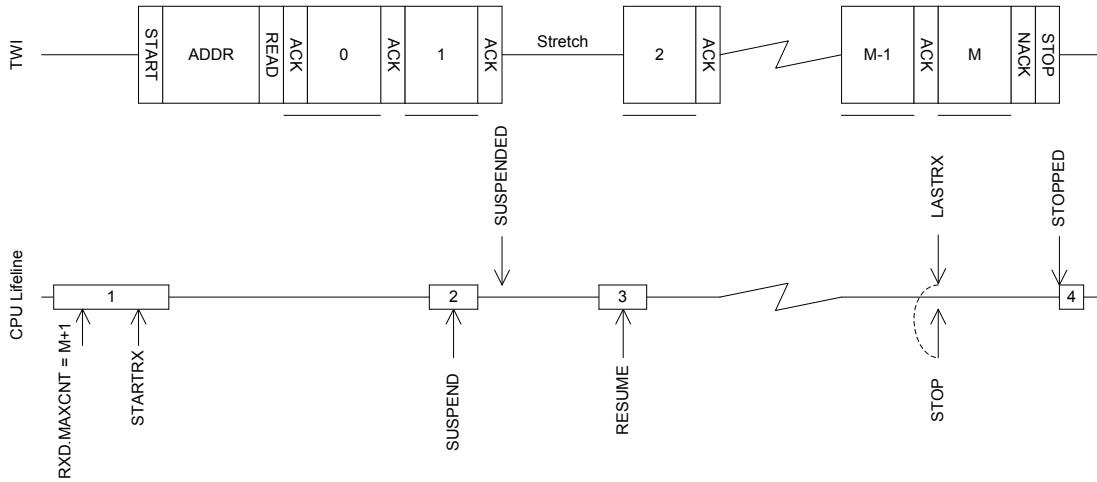


Figure 81: The TWI master reading data from a slave

### 33.5 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave. This example uses shortcuts to perform the simplest type of repeated start sequence, i.e. one write followed by one read. The same approach can be used to perform a repeated start sequence where the sequence is read followed by write.

The figure [Figure 82: A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave](#) on page 308 illustrates this:

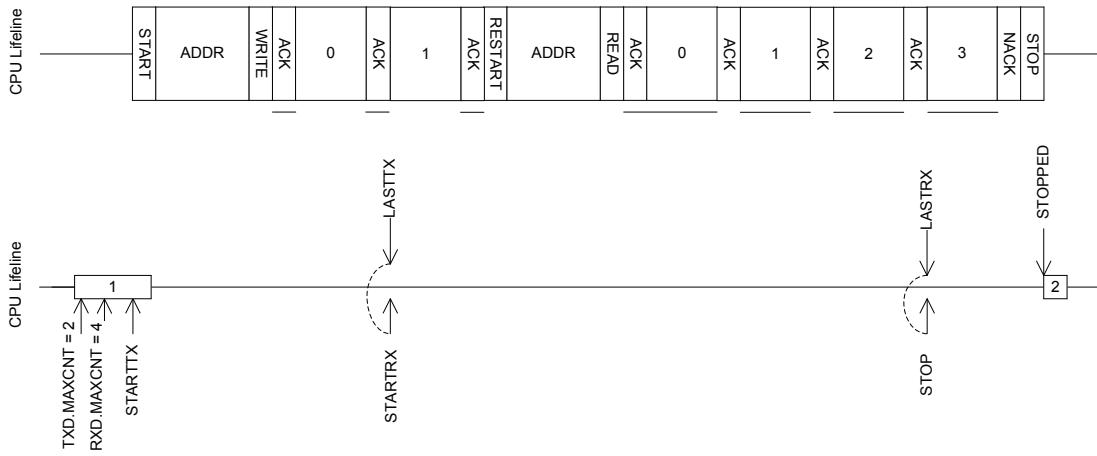
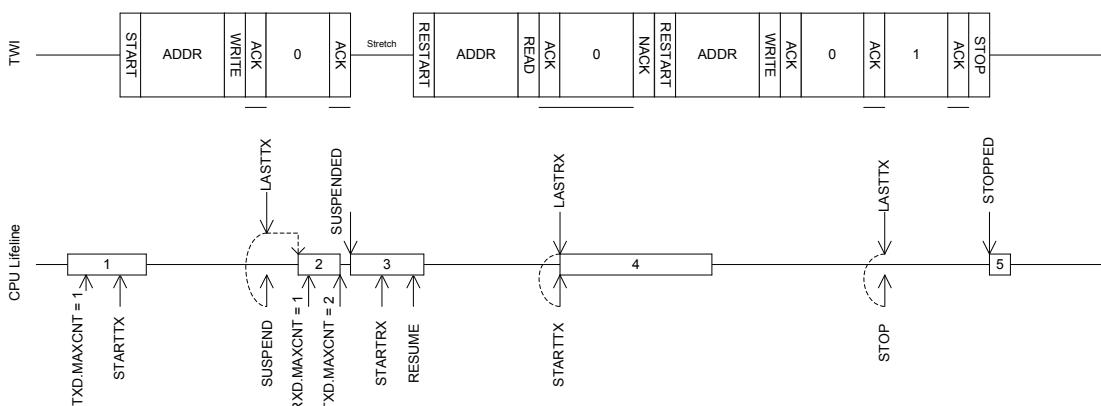


Figure 82: A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave

If a more complex repeated start sequence is needed and the TWI firmware drive is serviced in a low priority interrupt it may be necessary to use the SUSPEND task and SUSPENDED event to guarantee that the correct tasks are generated at the correct time. This is illustrated in [Figure 83: A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts](#) on page 309.



**Figure 83: A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts**

### 33.6 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

### 33.7 Master mode pin configuration

The SCL and SDA signals associated with the TWI master are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI master is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN\_CNF[n] register. PSEL.SCL, PSEL.SDA must only be configured when the TWI master is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in [Table 73: GPIO configuration before enabling peripheral](#) on page 309.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 73: GPIO configuration before enabling peripheral**

TWI master signal	TWI master pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

### 33.8 Registers

**Table 74: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWIM	TWIM0	Two-wire interface master 0	
0x40004000	TWIM	TWIM1	Two-wire interface master 1	

**Table 75: Register Overview**

Register	Offset	Description
TASKS_STARTRX	0x000	Start TWI receive sequence
TASKS_STARTTX	0x008	Start TWI transmit sequence
TASKS_STOP	0x014	Stop TWI transaction. Must be issued while the TWI master is not suspended.
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x124	TWI error
EVENTS_SUSPENDED	0x148	Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
EVENTS_RXSTARTED	0x14C	Receive sequence started
EVENTS_TXSTARTED	0x150	Transmit sequence started
EVENTS_LASTRX	0x15C	Byte boundary, starting to receive the last byte
EVENTS_LASTTX	0x160	Byte boundary, starting to transmit the last byte
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSRC	0x4C4	Error source
ENABLE	0x500	Enable TWIM
PSEL.SCL	0x508	Pin select for SCL signal
PSEL.SDA	0x50C	Pin select for SDA signal
FREQUENCY	0x524	TWI frequency
RXD.PTR	0x534	Data pointer
RXD.MAXCNT	0x538	Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C	Number of bytes transferred in the last transaction
RXD.LIST	0x540	EasyDMA list type
TXD.PTR	0x544	Data pointer
TXD.MAXCNT	0x548	Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C	Number of bytes transferred in the last transaction
TXD.LIST	0x550	EasyDMA list type
ADDRESS	0x588	Address used in the TWI transfer

### 33.8.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																							F	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	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																										
A	RW	LASTTX_STARTRX				Shortcut between LASTTX event and STARTRX task																										
						See <a href="#">EVENTS_LASTTX</a> and <a href="#">TASKS_STARTRX</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										
B	RW	LASTTX_SUSPEND				Shortcut between LASTTX event and SUSPEND task																										
						See <a href="#">EVENTS_LASTTX</a> and <a href="#">TASKS_SUSPEND</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										
C	RW	LASTTX_STOP				Shortcut between LASTTX event and STOP task																										
						See <a href="#">EVENTS_LASTTX</a> and <a href="#">TASKS_STOP</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										
D	RW	LASTRX_STARTTX				Shortcut between LASTRX event and STARTTX task																										
						See <a href="#">EVENTS_LASTRX</a> and <a href="#">TASKS_STARTTX</a>																										
		Disabled	0			Disable shortcut																										

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Id	F D C B A					
<b>Reset 0x00000000</b>	0 0					
Id	RW	Field	Value	Id	Value	Description
F	RW	LASTRX_STOP	Enabled	1		Enable shortcut
						Shortcut between LASTRX event and STOP task
						See <a href="#">EVENTS_LASTRX</a> and <a href="#">TASKS_STOP</a>
			Disabled	0		Disable shortcut
			Enabled	1		Enable shortcut

### 33.8.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Id	J I H G F D A					
<b>Reset 0x00000000</b>	0 0					
Id	RW	Field	Value	Id	Value	Description
A	RW	STOPPED	Enabled	1		Enable or disable interrupt for STOPPED event
						See <a href="#">EVENTS_STOPPED</a>
			Disabled	0		Disable
			Enabled	1		Enable
D	RW	ERROR	Enabled	1		Enable or disable interrupt for ERROR event
						See <a href="#">EVENTS_ERROR</a>
			Disabled	0		Disable
			Enabled	1		Enable
F	RW	SUSPENDED	Enabled	1		Enable or disable interrupt for SUSPENDED event
						See <a href="#">EVENTS_SUSPENDED</a>
			Disabled	0		Disable
			Enabled	1		Enable
G	RW	RXSTARTED	Enabled	1		Enable or disable interrupt for RXSTARTED event
						See <a href="#">EVENTS_RXSTARTED</a>
			Disabled	0		Disable
			Enabled	1		Enable
H	RW	TXSTARTED	Enabled	1		Enable or disable interrupt for TXSTARTED event
						See <a href="#">EVENTS_TXSTARTED</a>
			Disabled	0		Disable
			Enabled	1		Enable
I	RW	LASTRX	Enabled	1		Enable or disable interrupt for LASTRX event
						See <a href="#">EVENTS_LASTRX</a>
			Disabled	0		Disable
			Enabled	1		Enable
J	RW	LASTTX	Enabled	1		Enable or disable interrupt for LASTTX event
						See <a href="#">EVENTS_LASTTX</a>
			Disabled	0		Disable
			Enabled	1		Enable

### 33.8.3 INTENSET

Address offset: 0x304

Enable interrupt

### **33.8.4 INTENCLR**

Address offset: 0x308

Disable interrupt

### **33.8.5 ERRORSRC**

Address offset: 0x4C4

### Error source

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										C	B	A				
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value													Description															
A	RW	OVERRUN															Overrun error															
																	A new byte was received before previous byte got transferred into RXD buffer. (Previous data is lost)															
			NotReceived	0													Error did not occur															
			Received	1													Error occurred															
B	RW	ANACK															NACK received after sending the address (write '1' to clear)															
			NotReceived	0													Error did not occur															
			Received	1													Error occurred															
C	RW	DNACK															NACK received after sending a data byte (write '1' to clear)															
			NotReceived	0													Error did not occur															
			Received	1													Error occurred															

### **33.8.6 ENABLE**

Address offset: 0x500

## Enable TWIM

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

### **33.8.7 PSEL.SCL**

Address offset: 0x508

Pin select for SCL signal

### **33.8.8 PSEL.SDA**

Address offset: 0x50C

Pin select for SDA signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
		Disconnected	1		Disconnect																											
		Connected	0		Connect																											

### **33.8.9 FREQUENCY**

Address offset: 0x524

## TWI frequency

### **33.8.10 RXD.PTR**

Address offset: 0x534

## Data pointer

### **33.8.11 RXD.MAXCNT**

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	MAXCNT	[1..255]		Maximum number of bytes in receive buffer																											

### **33.8.12 RXD.AMOUNT**

Address offset: 0x53C

Number of bytes transferred in the last transaction

### **33.8.13 RXD.LIST**

Address offset: 0x540

## EasyDMA list type

33.8.14 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PTR			Data pointer																											

### **33.8.15 TXD.MAXCNT**

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	MAXCNT		[1..255]	Maximum number of bytes in transmit buffer

### 33.8.16 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

### 33.8.17 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	LIST			List type
		Disabled	0		Disable EasyDMA list
		ArrayList	1		Use array list

### 33.8.18 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

## 33.9 Electrical specification

### 33.9.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM}$	Bit rates for TWIM <sup>30</sup>	100	400		kbps
$I_{TWIM,100\text{ kbps}}$	Run current for TWIM, 100 kbps	50			$\mu\text{A}$
$I_{TWIM,400\text{ kbps}}$	Run current for TWIM, 400 kbps	50			$\mu\text{A}$
$t_{TWIM,START,LP}$	Time from STARTRX/STARTTX task to transmission started, Low power mode		$t_{TWIM,START}$		$\mu\text{s}$
			+		
			$t_{START\_HFIN}$		
$t_{TWIM,START,CL}$	Time from STARTRX/STARTTX task to transmission started, Constant latency mode	1.5			$\mu\text{s}$

<sup>30</sup> Higher bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

### 33.9.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL,100\text{kbps}}$	SCL clock frequency, 100 kbps	100			kHz
$f_{TWIM,SCL,250\text{kbps}}$	SCL clock frequency, 250 kbps	250			kHz
$f_{TWIM,SCL,400\text{kbps}}$	SCL clock frequency, 400 kbps	400			kHz
$t_{TWIM,SU\_DAT}$	Data setup time before positive edge on SCL – all modes	300			ns
$t_{TWIM,HD\_DAT}$	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIM,HD\_STA,100\text{kbps}}$	TWIM master hold time for START and repeated START condition, 100 kbps	10000			ns
$t_{TWIM,HD\_STA,250\text{kbps}}$	TWIM master hold time for START and repeated START condition, 250 kbps	4000			ns
$t_{TWIM,HD\_STA,400\text{kbps}}$	TWIM master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWIM,SU\_STO,100\text{kbps}}$	TWIM master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWIM,SU\_STO,250\text{kbps}}$	TWIM master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWIM,SU\_STO,400\text{kbps}}$	TWIM master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWIM,BUF,100\text{kbps}}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWIM,BUF,250\text{kbps}}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2700			ns
$t_{TWIM,BUF,400\text{kbps}}$	TWIM master bus free time between STOP and START conditions, 400 kbps	2100			ns

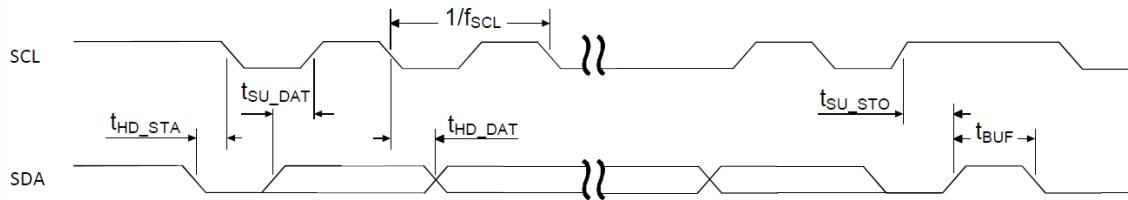


Figure 84: TWIM timing diagram, 1 byte transaction

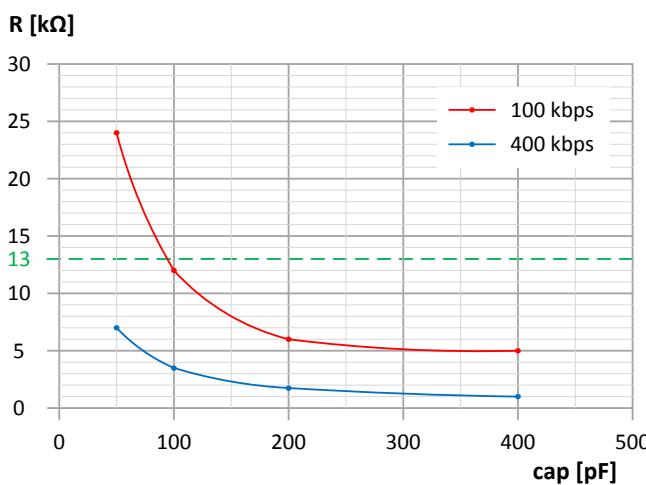
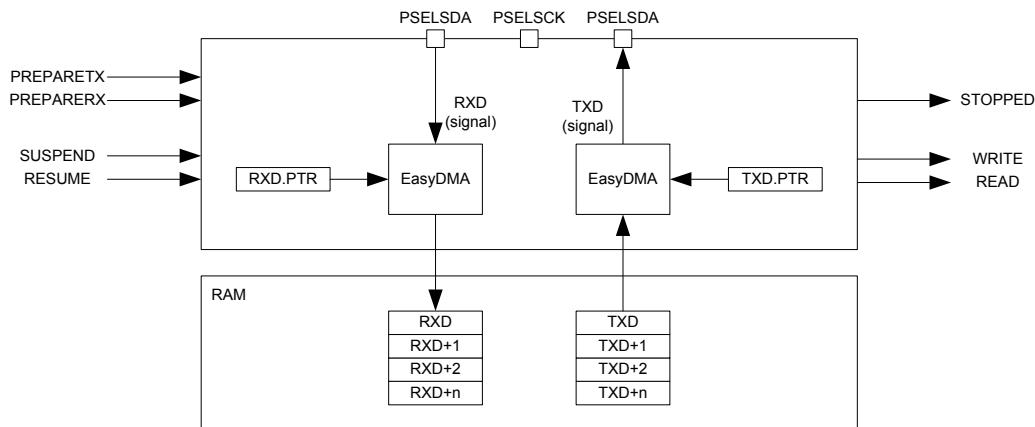


Figure 85: Recommended TWIM pullup value vs. line capacitance

- The I<sup>2</sup>C specification allows a line capacitance of 400 pF at most.
- The nRF52832 internal pullup has a fixed value of typ. 13 kOhm, see  $R_{PU}$  in the GPIO chapter.

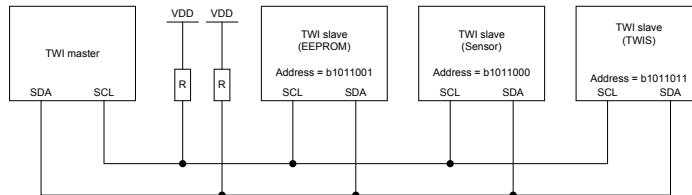
## 34 TWIS — I<sup>2</sup>C compatible two-wire interface slave with EasyDMA

TWI slave with EasyDMA (TWIS) is compatible with I<sup>2</sup>C operating at 100 kHz and 400 kHz. The TWI transmitter and receiver implement EasyDMA.



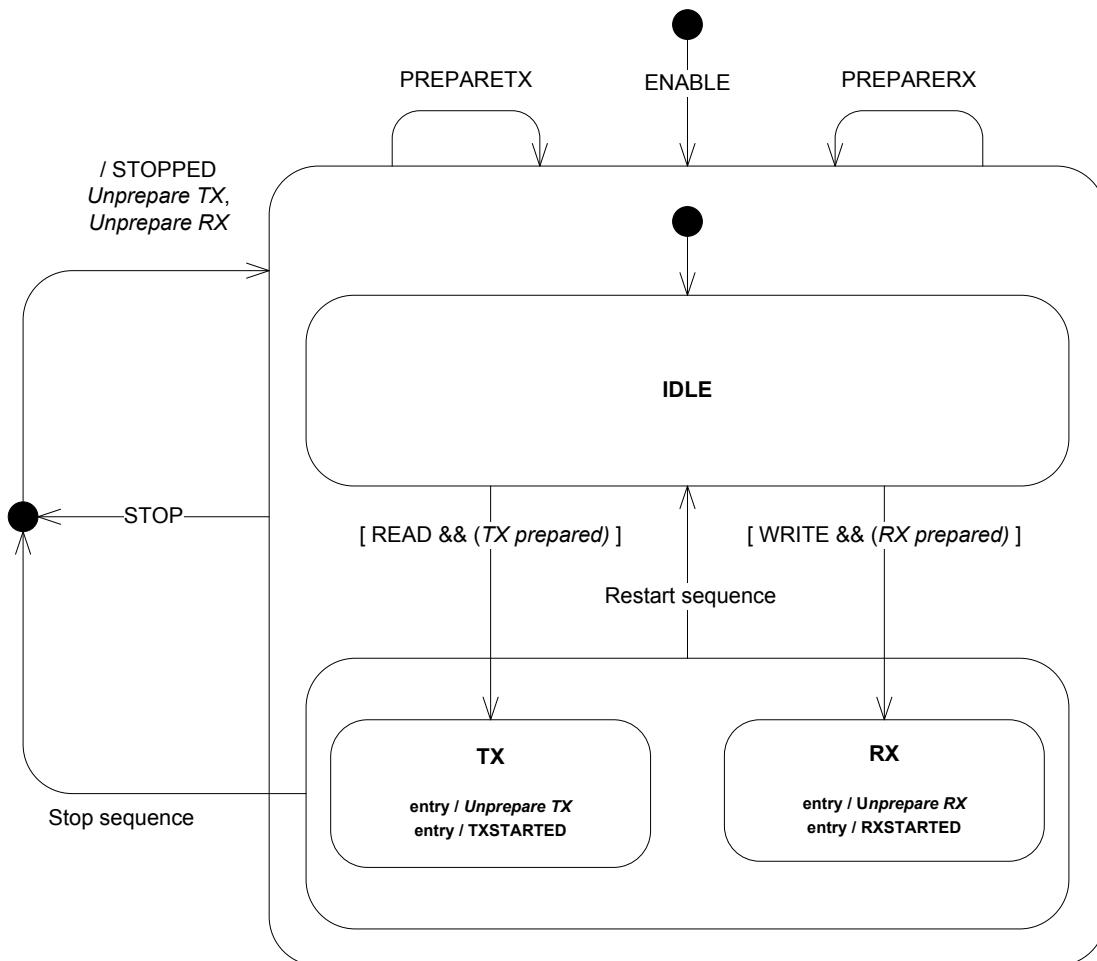
**Figure 86: TWI slave with EasyDMA**

A typical TWI setup consists of one master and one or more slaves. For an example, see [Figure 87: A typical TWI setup comprising one master and three slaves](#) on page 318. TWIS is only able to operate with a single master on the TWI bus.



**Figure 87: A typical TWI setup comprising one master and three slaves**

The TWI slave state machine is illustrated in [Figure 88: TWI slave state machine](#) on page 319 and [Table 76: TWI slave state machine symbols](#) on page 319 is explaining the different symbols used in the state machine.



**Figure 88: TWI slave state machine**

**Table 76: TWI slave state machine symbols**

Symbol	Type	Description
ENABLE	Register	The TWI slave has been enabled via the <a href="#">ENABLE</a> register
PREPARETX	Task	The <a href="#">TASKS_PREPARETX</a> task has been triggered
STOP	Task	The <a href="#">TASKS_STOP</a> task has been triggered
PREPARERX	Task	The <a href="#">TASKS_PREPARERX</a> task has been triggered
STOPPED	Event	The <a href="#">EVENTS_STOPPED</a> event was generated
RXSTARTED	Event	The <a href="#">EVENTS_RXSTARTED</a> event was generated
TXSTARTED	Event	The <a href="#">EVENTS_TXSTARTED</a> event was generated
TX prepared	Internal	Internal flag indicating that a <a href="#">TASKS_PREPARETX</a> task has been triggered. This flag is not visible to the user.
RX prepared	Internal	Internal flag indicating that a <a href="#">TASKS_PREPARERX</a> task has been triggered. This flag is not visible to the user.
Unprepare TX	Internal	Clears the internal 'TX prepared' flag until next <a href="#">TASKS_PREPARETX</a> task.
Unprepare RX	Internal	Clears the internal 'RX prepared' flag until next <a href="#">TASKS_PREPARERX</a> task.
Stop sequence	TWI protocol	A TWI stop sequence was detected
Restart sequence	TWI protocol	A TWI restart sequence was detected

The TWI slave supports clock stretching performed by the master.

The TWI slave operates in a low power mode while waiting for a TWI master to initiate a transfer. As long as the TWI slave is not addressed, it will remain in this low power mode.

To secure correct behaviour of the TWI slave, PSEL.SCL, PSEL.SDA, CONFIG and the ADDRESS[n] registers, must be configured prior to enabling the TWI slave through the ENABLE register. Similarly, changing these settings must be performed while the TWI slave is disabled. Failing to do so may result in unpredictable behaviour.

## 34.1 Shared resources

The TWI slave shares registers and other resources with other peripherals that have the same ID as the TWI slave.

Therefore, you must disable all peripherals that have the same ID as the TWI slave before the TWI slave can be configured and used. Disabling a peripheral that has the same ID as the TWI slave will not reset any of the registers that are shared with the TWI slave. It is therefore important to configure all relevant registers explicitly to secure that the TWI slave operates correctly.

The Instantiation table in [Instantiation](#) on page 24 shows which peripherals have the same ID as the TWI slave.

## 34.2 EasyDMA

The TWI slave implements EasyDMA for reading and writing to and from the RAM.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

## 34.3 TWI slave responding to a read command

Before the TWI slave can respond to a read command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume  $I_{IDLE}$ .

A read command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the TWI slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the read command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a READ event when it acknowledges the read command.

The TWI slave is only able to detect a read command from the IDLE state.

The TWI slave will set an internal 'TX prepared' flag when the PREPARETX task is triggered.

When the read command is received the TWI slave will enter the TX state if the internal 'TX prepared' flag is set.

If the internal 'TX prepared' flag is not set when the read command is received, the TWI slave will stretch the master's clock until the PREPARETX task is triggered and the internal 'TX prepared' flag is set.

The TWI slave will generate the TXSTARTED event and clear the 'TX prepared' flag ('unprepare TX') when it enters the TX state. In this state the TWI slave will send the data bytes found in the transmit buffer to the master using the master's clock. The TWI slave will consume  $I_{TX}$  in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the TX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the 'TX prepared' flag ('unprepare TX') and go back to the IDLE state when it has stopped.

The transmit buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to send TXD.MAXCNT bytes from the transmit buffer for each transaction. If the TWI master

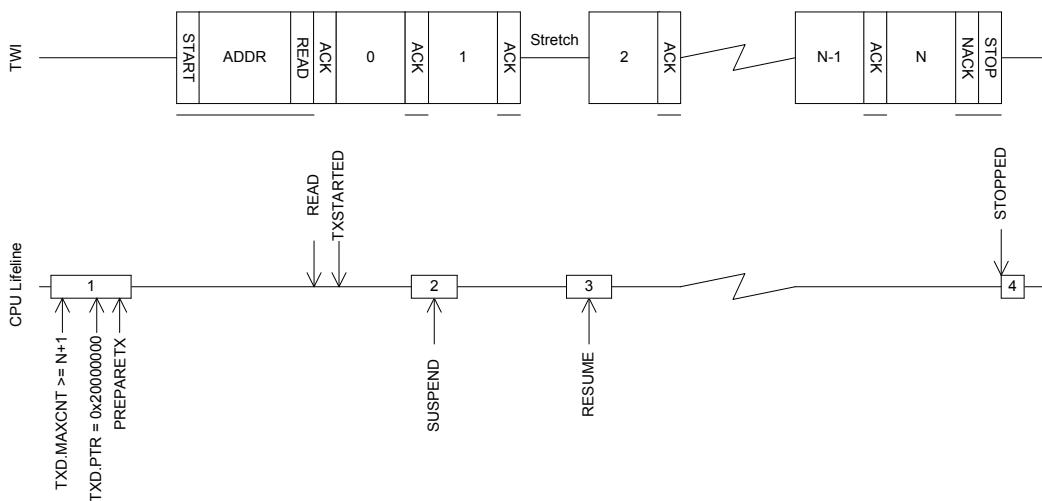
forces the slave to send more than TXD.MAXCNT bytes, the slave will send the byte specified in the ORC register to the master instead. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see TXD.PTR etc., are latched when the TXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the 'TX prepared' flag and go back to the IDLE state when it has stopped, see also [Terminating an ongoing TWI transaction](#) on page 323.

Each byte sent from the slave will be followed by an ACK/NACK bit sent from the master. The TWI master will generate a NACK following the last byte that it wants to receive to tell the slave to release the bus so that the TWI master can generate the stop condition. The TXD.AMOUNT register can be queried after a transaction to see how many bytes were sent.

A typical TWI slave read command response is illustrated in [Figure 89: The TWI slave responding to a read command](#) on page 321. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.



**Figure 89: The TWI slave responding to a read command**

### 34.4 TWI slave responding to a write command

Before the TWI slave can respond to a write command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume  $I_{IDLE}$ .

A write command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the write command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a WRITE event if it acknowledges the write command.

The TWI slave is only able to detect a write command from the IDLE state.

The TWI slave will set an internal 'RX prepared' flag when the PREPARERX task is triggered.

When the write command is received the TWI slave will enter the RX state if the internal 'RX prepared' flag is set.

If the internal 'RX prepared' flag is not set when the write command is received, the TWI slave will stretch the master's clock until the PREPARERX task is triggered and the internal 'RX prepared' flag is set.

The TWI slave will generate the RXSTARTED event and clear the internal 'RX prepared' flag ('unprepare RX') when it enters the RX state. In this state the TWI slave will be able to receive the bytes sent by the TWI master. The TWI slave will consume  $I_{RX}$  in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the RX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the internal 'RX prepared' flag ('unprepare RX') and go back to the IDLE state when it has stopped.

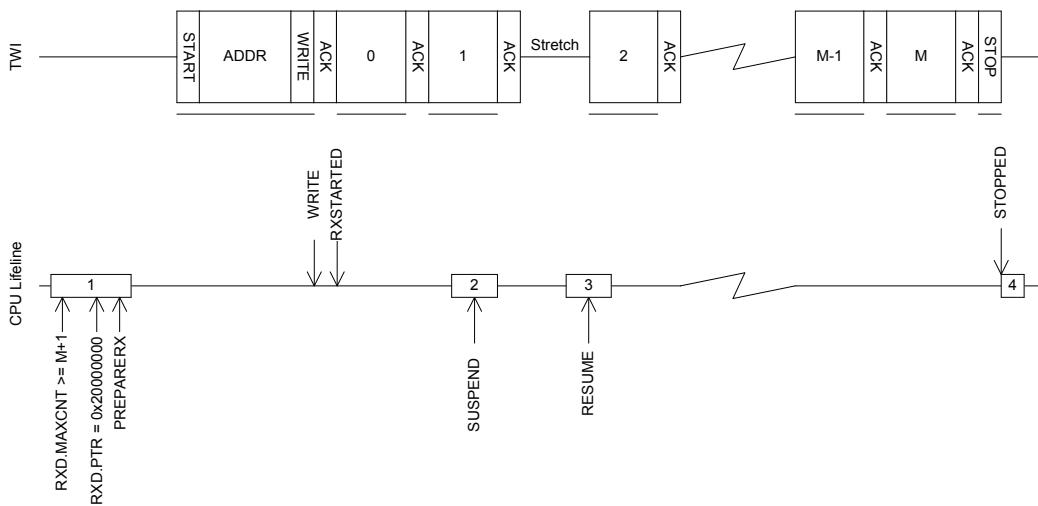
The receive buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to receive as many bytes as specified in the RXD.MAXCNT register. If the TWI master tries to send more bytes to the slave than the slave is able to receive, these bytes will be discarded and the bytes will be NACKed by the slave. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see RXD.PTR etc., are latched when the RXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the internal 'RX prepared' flag and go back to the IDLE state when it has stopped, see also [Terminating an ongoing TWI transaction](#) on page 323.

The TWI slave will generate an ACK after every byte received from the master. The RXD.AMOUNT register can be queried after a transaction to see how many bytes were received.

A typical TWI slave write command response is illustrated in [Figure 90: The TWI slave responding to a write command](#) on page 322. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.



**Figure 90: The TWI slave responding to a write command**

## 34.5 Master repeated start sequence

An example of a repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave.

This is illustrated in [Figure 91: A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave](#) on page 323.

It is here assumed that the receiver does not know in advance what the master wants to read, and that this information is provided in the first two bytes received in the write part of the repeated start sequence. To guarantee that the CPU is able to process the received data before the TWI slave starts to reply to the read command, the SUSPEND task is triggered via a shortcut from the READ event generated when the read command is received. When the CPU has processed the incoming data and prepared the correct data response, the CPU will resume the transaction by triggering the RESUME task.

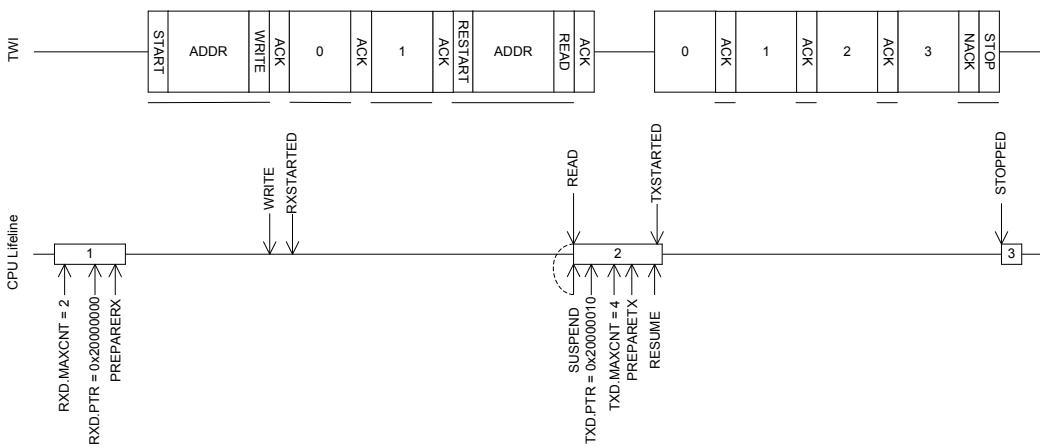


Figure 91: A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave

## 34.6 Terminating an ongoing TWI transaction

In some situations, e.g. if the external TWI master is not responding correctly, it may be required to terminate an ongoing transaction.

This can be achieved by triggering the STOP task. In this situation a STOPPED event will be generated when the TWI has stopped independent of whether or not a STOP condition has been generated on the TWI bus. The TWI slave will release the bus when it has stopped and go back to its IDLE state.

## 34.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

## 34.8 Slave mode pin configuration

The SCL and SDA signals associated with the TWI slave are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI slave is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN\_CNF[n] register. PSEL.SCL and PSEL.SDA must only be configured when the TWI slave is disabled.

To secure correct signal levels on the pins used by the TWI slave when the system is in OFF mode, and when the TWI slave is disabled, these pins must be configured in the GPIO peripheral as described in [Table 77: GPIO configuration before enabling peripheral](#) on page 323.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 77: GPIO configuration before enabling peripheral**

TWI slave signal	TWI slave pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

## 34.9 Registers

**Table 78: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWIS	TWISO	Two-wire interface slave 0	
0x40004000	TWIS	TWIS1	Two-wire interface slave 1	

**Table 79: Register Overview**

Register	Offset	Description
TASKS_STOP	0x014	Stop TWI transaction
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction
TASKS_PREPARERX	0x030	Prepare the TWI slave to respond to a write command
TASKS_PREPARETX	0x034	Prepare the TWI slave to respond to a read command
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x124	TWI error
EVENTS_RXSTARTED	0x14C	Receive sequence started
EVENTS_TXSTARTED	0x150	Transmit sequence started
EVENTS_WRITE	0x164	Write command received
EVENTS_READ	0x168	Read command received
<i>SHORTS</i>	0x200	Shortcut register
<i>INTEN</i>	0x300	Enable or disable interrupt
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>ERRORSRC</i>	0x4D0	Error source
<i>MATCH</i>	0x4D4	Status register indicating which address had a match
<i>ENABLE</i>	0x500	Enable TWIS
<i>PSEL.SCL</i>	0x508	Pin select for SCL signal
<i>PSEL.SDA</i>	0x50C	Pin select for SDA signal
<i>RXD.PTR</i>	0x534	RXD Data pointer
<i>RXD.MAXCNT</i>	0x538	Maximum number of bytes in RXD buffer
<i>RXD.AMOUNT</i>	0x53C	Number of bytes transferred in the last RXD transaction
<i>TXD.PTR</i>	0x544	TXD Data pointer
<i>TXD.MAXCNT</i>	0x548	Maximum number of bytes in TXD buffer
<i>TXD.AMOUNT</i>	0x54C	Number of bytes transferred in the last TXD transaction
<i>ADDRESS[0]</i>	0x588	TWI slave address 0
<i>ADDRESS[1]</i>	0x58C	TWI slave address 1
<i>CONFIG</i>	0x594	Configuration register for the address match mechanism
<i>ORC</i>	0x5C0	Over-read character. Character sent out in case of an over-read of the transmit buffer.

### 34.9.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	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	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	WRITE_SUSPEND				Shortcut between WRITE event and SUSPEND task																										
						See <a href="#">EVENTS_WRITE</a> and <a href="#">TASKS_SUSPEND</a>																										
				Disabled	0	Disable shortcut																										
				Enabled	1	Enable shortcut																										
B	RW	READ_SUSPEND				Shortcut between READ event and SUSPEND task																										
						See <a href="#">EVENTS_READ</a> and <a href="#">TASKS_SUSPEND</a>																										

Bit number	31	30	29	28	27	26	25	24	23	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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value	Description																											
			Disabled	0	Disable shortcut																											
			Enabled	1	Enable shortcut																											

## 34.9.2 INTEN

Address offset: 0x300

## Enable or disable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1				
Id			H	G		F	E		B		A																										
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>			<b>Value</b>			<b>Description</b>																												
A	RW	STOPPED				Enable or disable interrupt for STOPPED event																															
						See <a href="#">EVENTS_STOPPED</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											
B	RW	ERROR				Enable or disable interrupt for ERROR event																															
						See <a href="#">EVENTS_ERROR</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											
E	RW	RXSTARTED				Enable or disable interrupt for RXSTARTED event																															
						See <a href="#">EVENTS_RXSTARTED</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											
F	RW	TXSTARTED				Enable or disable interrupt for TXSTARTED event																															
						See <a href="#">EVENTS_TXSTARTED</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											
G	RW	WRITE				Enable or disable interrupt for WRITE event																															
						See <a href="#">EVENTS_WRITE</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											
H	RW	READ				Enable or disable interrupt for READ event																															
						See <a href="#">EVENTS_READ</a>																															
						Disabled	0			Disable																											
						Enabled	1			Enable																											

### **34.9.3 INTENSET**

Address offset: 0x304

## Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id								H	G			F	E									B								A		
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id																								Description					
A	RW	STOPPED																									Write '1' to Enable interrupt for STOPPED event					
																											See <a href="#">EVENTS_STOPPED</a>					
																											Set					
																											1					
																											Enable					
																											Disabled					
																											0					
																											Read: Disabled					
																											Enabled					
																											1					
																											Read: Enabled					
B	RW	ERROR																									Write '1' to Enable interrupt for ERROR event					

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			H	G		F	E		B		A																							
<b>Reset 0x00000000</b>			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																												
						See <a href="#">EVENTS_ERROR</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>E</b>	<b>RW</b>	<b>RXSTARTED</b>				Write '1' to Enable interrupt for RXSTARTED event																												
						See <a href="#">EVENTS_RXSTARTED</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>F</b>	<b>RW</b>	<b>TXSTARTED</b>				Write '1' to Enable interrupt for TXSTARTED event																												
						See <a href="#">EVENTS_TXSTARTED</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>G</b>	<b>RW</b>	<b>WRITE</b>				Write '1' to Enable interrupt for WRITE event																												
						See <a href="#">EVENTS_WRITE</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>H</b>	<b>RW</b>	<b>READ</b>				Write '1' to Enable interrupt for READ event																												
						See <a href="#">EVENTS_READ</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>B</b>	<b>RW</b>	<b>ERROR</b>				Write '1' to Disable interrupt for ERROR event																												
						See <a href="#">EVENTS_ERROR</a>																												
		Clear	1			Disable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>E</b>	<b>RW</b>	<b>RXSTARTED</b>				Write '1' to Disable interrupt for RXSTARTED event																												
						See <a href="#">EVENTS_RXSTARTED</a>																												
		Clear	1			Disable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
<b>F</b>	<b>RW</b>	<b>TXSTARTED</b>				Write '1' to Disable interrupt for TXSTARTED event																												
						See <a href="#">EVENTS_TXSTARTED</a>																												
		Clear	1			Disable																												

### **34.9.5 ERRORSRC**

Address offset: 0x4D0

### Error source

## **34.9.6 MATCH**

Address offset: 0x4D4

Status register indicating which address had a match

### **34.9.7 ENABLE**

Address offset: 0x500

## Enable TWIS

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

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 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				
Id	RW	Field	Value Id	Value	Description
A	RW	PIN	[0..31]		Pin number
B	RW	CONNECT			Connection
		Disconnected	1		Disconnect
		Connected	0		Connect

### 34.9.8 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	B				
Reset 0xFFFFFFFF	1 1				
Id	RW	Field	Value Id	Value	Description
A	RW	PIN	[0..31]		Pin number
B	RW	CONNECT			Connection
		Disconnected	1		Disconnect
		Connected	0		Connect

### 34.9.9 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	B				
Reset 0xFFFFFFFF	1 1				
Id	RW	Field	Value Id	Value	Description
A	RW	PIN	[0..31]		Pin number
B	RW	CONNECT			Connection
		Disconnected	1		Disconnect
		Connected	0		Connect

### 34.9.10 RXD.PTR

Address offset: 0x534

RXD Data pointer

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

### 34.9.11 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in RXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	MAXCNT			Maximum number of bytes in RXD buffer

### 34.9.12 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last RXD transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	AMOUNT			Number of bytes transferred in the last RXD transaction																											

## **34.9.13 TXD.PTR**

Address offset: 0x544

## TXD Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	RW	PTR				TXD Data pointer																										

### 34.9.14 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in TXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	MAXCNT			Maximum number of bytes in TXD buffer

### **34.9.15 TXD.AMOUNT**

Address offset: 0x54C

Number of bytes transferred in the last TXD transaction

### 34.9.16 ADDRESS[0]

Address offset: 0x588

## TWI slave 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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																											
A	RW	ADDRESS				TWI slave address																											

### **34.9.17 ADDRESS[1]**

Address offset: 0x58C

## TWI slave 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	ADDRESS			TWI slave address																											

### 34.9.18 CONFIG

Address offset: 0x594

Configuration register for the address match mechanism

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																															B	A
<b>Reset 0x00000001</b>	0	0	0	0	0	0	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	ADDRESS0				Enable or disable address matching on ADDRESS[0]																										
		Disabled	0			Disabled																										
		Enabled	1			Enabled																										
B	RW	ADDRESS1				Enable or disable address matching on ADDRESS[1]																										
		Disabled	0			Disabled																										
		Enabled	1			Enabled																										

### 34.9.19 ORC

Address offset: 0x5C0

Over-read character. Character sent out in case of an over-read of the transmit buffer.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																															A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	ORC				Over-read character. Character sent out in case of an over-read of the transmit buffer.																										

## 34.10 Electrical specification

### 34.10.1 TWIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f <sub>TWIS</sub>	Bit rates for TWIS <sup>31</sup>	100	400		kbps
I <sub>TWIS,100kbps</sub>	Run current for TWIS (Average current to receive and transfer a byte to RAM), 100 kbps		45		µA
I <sub>TWIS,400kbps</sub>	Run current for TWIS (Average current to receive and transfer a byte to RAM), 400 kbps		45		µA
I <sub>TWIS,IDL</sub>	Idle current for TWIS	1			µA
t <sub>TWIS,START,LP</sub>	Time from PREPARERX/PREPARETX task to ready to receive/transmit, Low power mode		t <sub>TWIS,START,</sub> + t <sub>START_HFIN</sub>		µs
t <sub>TWIS,START,CL</sub>	Time from PREPARERX/PREPARETX task to ready to receive/transmit, Constant latency mode	1.5			µs

### 34.10.2 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
f <sub>TWIS,SCL,400kbps</sub>	SCL clock frequency, 400 kbps		400		kHz
t <sub>TWIS,SU_DAT</sub>	Data setup time before positive edge on SCL – all modes	300			ns
t <sub>TWIS,HD_DAT</sub>	Data hold time after negative edge on SCL – all modes	500			ns
t <sub>TWIS,HD_STA,100kbps</sub>	TWI slave hold time from for START condition (SDA low to SCL low), 100 kbps	5200			ns
t <sub>TWIS,HD_STA,400kbps</sub>	TWI slave hold time from for START condition (SDA low to SCL low), 400 kbps	1300			ns

<sup>31</sup> Higher bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TWIS,SU,STO,100kbps}$	TWI slave setup time from SCL high to STOP condition, 100 kbps	5200			ns
$t_{TWIS,SU,STO,400kbps}$	TWI slave setup time from SCL high to STOP condition, 400 kbps	1300			ns
$t_{TWIS,BUF,100kbps}$	TWI slave bus free time between STOP and START conditions, 100 kbps		4700		ns
$t_{TWIS,BUF,400kbps}$	TWI slave bus free time between STOP and START conditions, 400 kbps	1300			ns

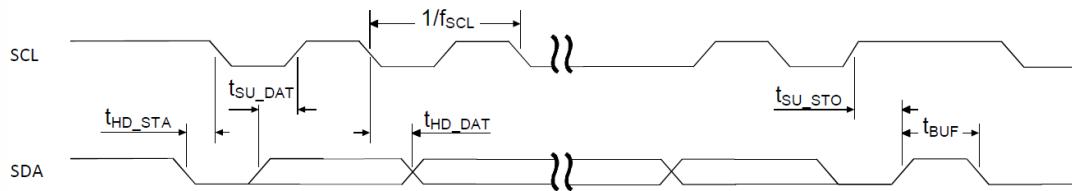


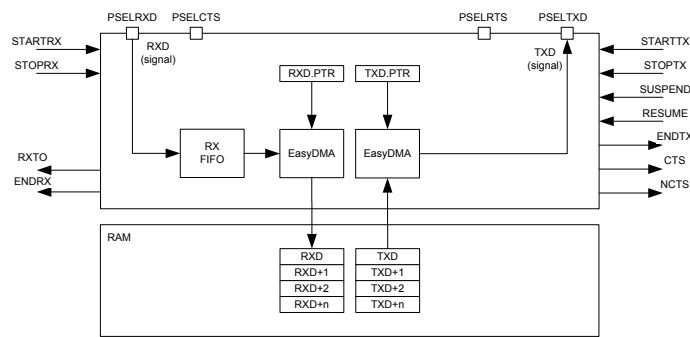
Figure 92: TWIS timing diagram, 1 byte transaction

# 35 UARTE — Universal asynchronous receiver/ transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA (UARTE) offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in hardware at a rate up to 1 Mbps, and EasyDMA data transfer from/to RAM.

Listed here are the main features for UARTE:

- Full-duplex operation
- Automatic hardware flow control
- Parity checking and generation for the 9<sup>th</sup> data bit
- EasyDMA
- Up to 1 Mbps baudrate
- Return to IDLE between transactions supported (when using HW flow control)
- One stop bit
- Least significant bit (LSB) first



**Figure 93: UARTE configuration**

The GPIOs used for each UART interface can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

## 35.1 Shared resources

The UARTE shares registers and other resources with other peripherals that have the same ID as the UARTE.

Therefore, you must disable all peripherals that have the same ID as the UARTE before the UARTE can be configured and used. Disabling a peripheral that has the same ID as the UARTE will not reset any of the registers that are shared with the UARTE. It is therefore important to configure all relevant UARTE registers explicitly to ensure that it operates correctly.

See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

## 35.2 EasyDMA

The UARTE implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The ENDRX/ENDTX event indicates that EasyDMA has finished accessing respectively the RX/TX buffer in RAM.

### 35.3 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to TXD.PTR, and the number of bytes in the RAM buffer to TXD.MAXCNT. The UARTE transmission is started by triggering the STARTTX task.

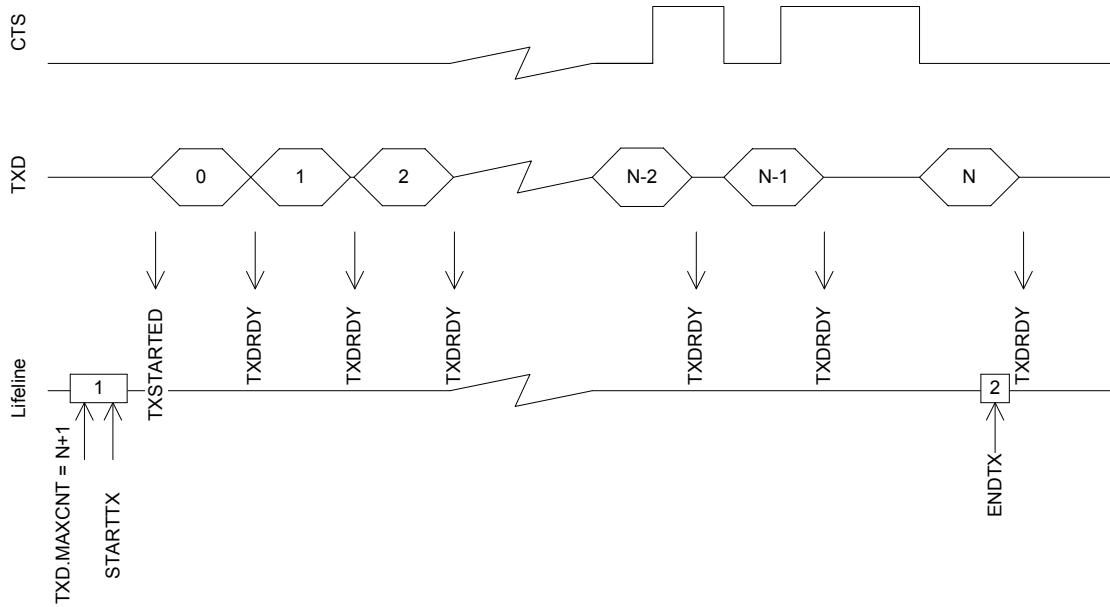
After each byte has been sent over the TXD line, a TXDRDY event will be generated.

When all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have been transmitted, the UARTE transmission will end automatically and an ENDTX event will be generated.

A UARTE transmission sequence is stopped by triggering the STOPTX task, a TXSTOPPED event will be generated when the UARTE transmitter has stopped.

If the ENDTX event has not already been generated when the UARTE transmitter has come to a stop, the UARTE will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

If flow control is enabled, a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in [Figure 94: UARTE transmission](#) on page 333. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended.



**Figure 94: UARTE transmission**

The UARTE transmitter will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTTX or after it has been stopped via STOPTX and the TXSTOPPED event has been generated. See [POWER — Power supply](#) on page 78 for more information about power modes.

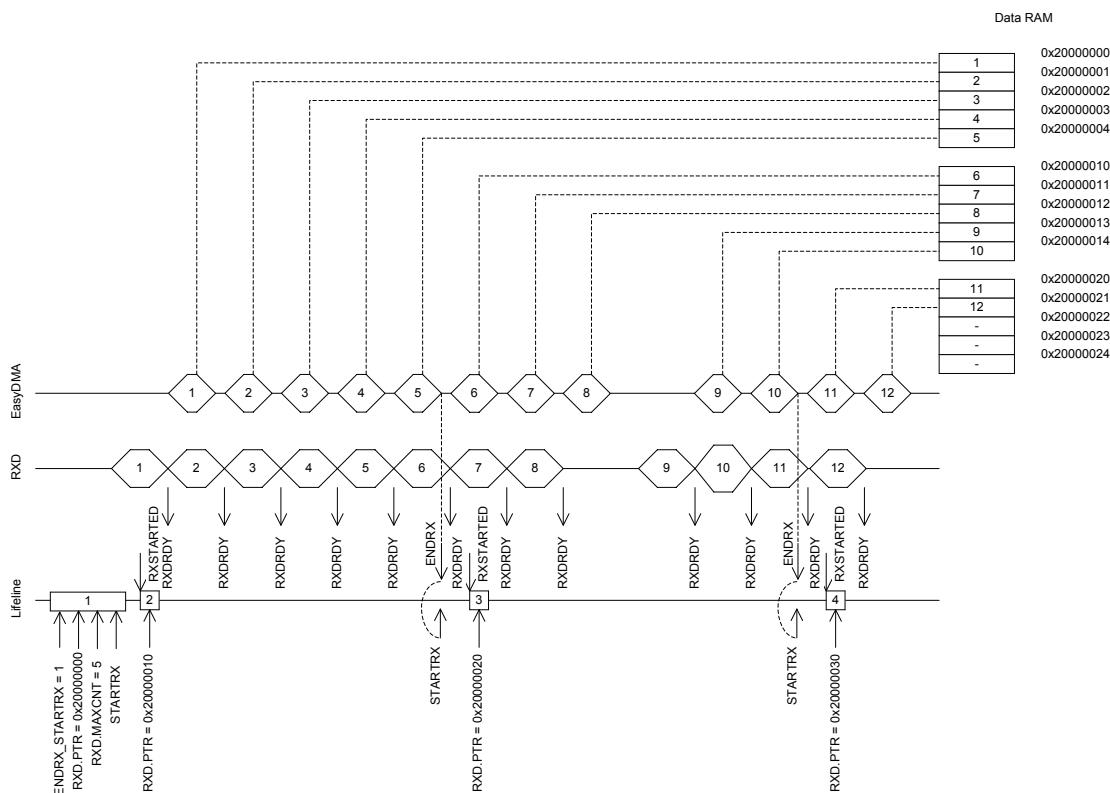
### 35.4 Reception

The UARTE receiver is started by triggering the STARTRX task. The UARTE receiver is using EasyDMA to store incoming data in an RX buffer in RAM.

The RX buffer is located at the address specified in the RXD.PTR register. The RXD.PTR register is double-buffered and it can be updated and prepared for the next STARTRX task immediately after the RXSTARTED event is generated. The size of the RX buffer is specified in the RXD.MAXCNT register and the UARTE will generate an ENDRX event when it has filled up the RX buffer, see [Figure 95: UARTE reception](#) on page 334.

For each byte received over the RXD line, an RXDRDY event will be generated. This event is likely to occur before the corresponding data has been transferred to Data RAM.

The RXD.AMOUNT register can be queried following an ENDRX event to see how many new bytes have been transferred to the RX buffer in RAM since the previous ENDRX event.



**Figure 95: UARTE reception**

The UARTE receiver is stopped by triggering the STOPRX task. An RXTO event is generated when the UARTE has stopped. The UARTE will make sure that an impending ENDRX event will be generated before the RXTO event is generated. This means that the UARTE will guarantee that no ENDRX event will be generated after RXTO, unless the UARTE is restarted or a FLUSHRX command is issued after the RXTO event is generated.

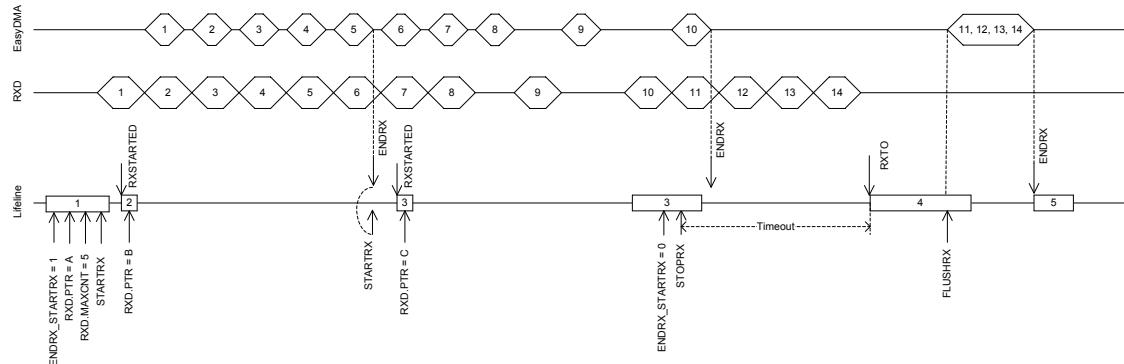
**Important:** If the ENDRX event has not already been generated when the UARTE receiver has come to a stop, which implies that all pending content in the RX FIFO has been moved to the RX buffer, the UARTE will generate the ENDRX event explicitly even though the RX buffer is not full. In this scenario the ENDRX event will be generated before the RXTO event is generated.

To be able to know how many bytes have actually been received into the RX buffer, the CPU can read the RXD.AMOUNT register following the ENDRX event or the RXTO event.

The UARTE is able to receive up to four bytes after the STOPRX task has been triggered as long as these are sent in succession immediately after the RTS signal is deactivated. This is possible because after the RTS is deactivated the UARTE is able to receive bytes for an extended period equal to the time it takes to send 4 bytes on the configured baud rate.

After the RXTO event is generated the internal RX FIFO may still contain data, and to move this data to RAM the FLUSHRX task must be triggered. To make sure that this data does not overwrite data in the RX buffer, the RX buffer should be emptied or the RXD.PTR should be updated before the FLUSHRX task is triggered.

To make sure that all data in the RX FIFO is moved to the RX buffer, the RXD.MAXCNT register must be set to RXD.MAXCNT > 4, see [Figure 96: UARTE reception with forced stop via STOPRX](#) on page 335. The UARTE will generate the ENDRX event after completing the FLUSHRX task even if the RX FIFO was empty or if the RX buffer does not get filled up. To be able to know how many bytes have actually been received into the RX buffer in this case, the CPU can read the RXD.AMOUNT register following the ENDRX event.



**Figure 96: UARTE reception with forced stop via STOPRX**

If HW flow control is enabled the RTS signal will be deactivated when the receiver is stopped via the STOPRX task or when the UARTE is only able to receive four more bytes in its internal RX FIFO.

With flow control disabled, the UARTE will function in the same way as when the flow control is enabled except that the RTS line will not be used. This means that no signal will be generated when the UARTE has reached the point where it is only able to receive four more bytes in its internal RX FIFO. Data received when the internal RX FIFO is filled up, will be lost.

The UARTE receiver will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTRX or after it has been stopped via STOPRX and the RXTO event has been generated. See [POWER — Power supply](#) on page 78 for more information about power modes.

## 35.5 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

An ERROR event will not stop reception. If the error was a parity error, the received byte will still be transferred into Data RAM, and so will following incoming bytes. If there was a framing error (wrong stop bit), that specific byte will NOT be stored into Data RAM, but following incoming bytes will.

### 35.6 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

## 35.7 Parity configuration

When parity is enabled, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

## 35.8 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOPTX and STOPRX tasks may not be always needed (the peripheral might already be stopped), but if STOPTX and/or STOPRX is sent, software shall wait until the TXSTOPPED and/or RXTO event is received in response, before disabling the peripheral through the ENABLE register.

## 35.9 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UARTE are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UARTE is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.RTS, PSEL.RTS and PSEL.TXD must only be configured when the UARTE is disabled.

To secure correct signal levels on the pins by the UARTE when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in [Table 80: GPIO configuration before enabling peripheral](#) on page 336.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 80: GPIO configuration before enabling peripheral**

UARTE signal	UARTE pin	Direction	Output value	Comments
RXD	As specified in PSEL.RXD	Input	Not applicable	
CTS	As specified in PSEL.CTS	Input	Not applicable	
RTS	As specified in PSEL.RTS	Output	1	
TXD	As specified in PSEL.TXD	Output	1	

## 35.10 Registers

**Table 81: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40002000	UARTE	UARTE0	Universal Asynchronous Receiver/ Transmitter with EasyDMA	

**Table 82: Register Overview**

Register	Offset	Description
TASKS_STARTRX	0x000	Start UART receiver
TASKS_STOPRX	0x004	Stop UART receiver
TASKS_STARTTX	0x008	Start UART transmitter
TASKS_STOPTX	0x00C	Stop UART transmitter
TASKS_FLUSHRX	0x02C	Flush RX FIFO into RX buffer
EVENTS_CTS	0x100	CTS is activated (set low). Clear To Send.
EVENTS_NCTS	0x104	CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108	Data received in RXD (but potentially not yet transferred to Data RAM)
EVENTS_ENDRX	0x110	Receive buffer is filled up
EVENTS_TXDRDY	0x11C	Data sent from TXD
EVENTS_ENDTX	0x120	Last TX byte transmitted
EVENTS_ERROR	0x124	Error detected
EVENTS_RXTO	0x144	Receiver timeout
EVENTS_RXSTARTED	0x14C	UART receiver has started
EVENTS_TXSTARTED	0x150	UART transmitter has started
EVENTS_TXSTOPPED	0x158	Transmitter stopped
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt

Register	Offset	Description
<i>ERRORSRC</i>	0x480	Error source
<i>ENABLE</i>	0x500	Enable UART
<i>PSEL.RTS</i>	0x508	Pin select for RTS signal
<i>PSEL.TXD</i>	0x50C	Pin select for TXD signal
<i>PSEL.CTS</i>	0x510	Pin select for CTS signal
<i>PSEL.RXD</i>	0x514	Pin select for RXD signal
<i>BAUDRATE</i>	0x524	Baud rate. Accuracy depends on the HFCLK source selected.
<i>RXD.PTR</i>	0x534	Data pointer
<i>RXD.MAXCNT</i>	0x538	Maximum number of bytes in receive buffer
<i>RXD.AMOUNT</i>	0x53C	Number of bytes transferred in the last transaction
<i>TXD.PTR</i>	0x544	Data pointer
<i>TXD.MAXCNT</i>	0x548	Maximum number of bytes in transmit buffer
<i>TXD.AMOUNT</i>	0x54C	Number of bytes transferred in the last transaction
<i>CONFIG</i>	0x56C	Configuration of parity and hardware flow control

### 35.10.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																														D	C	
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
C	RW	ENDRX_STARTRX				Shortcut between ENDRX event and STARTRX task																										
						See <a href="#">EVENTS_ENDRX</a> and <a href="#">TASKS_STARTRX</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										
D	RW	ENDRX_STOPRX				Shortcut between ENDRX event and STOPRX task																										
						See <a href="#">EVENTS_ENDRX</a> and <a href="#">TASKS_STOPRX</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										

### 35.10.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																		L	J	I	H		G	F	E	D	C	B	A			
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	CTS				Enable or disable interrupt for CTS event																										
						See <a href="#">EVENTS_CTS</a>																										
		Disabled	0			Disable																										
		Enabled	1			Enable																										
B	RW	NCTS				Enable or disable interrupt for NCTS event																										
						See <a href="#">EVENTS_NCTS</a>																										
		Disabled	0			Disable																										
		Enabled	1			Enable																										
C	RW	RXDRDY				Enable or disable interrupt for RXDRDY event																										
						See <a href="#">EVENTS_RXDRDY</a>																										
		Disabled	0			Disable																										
		Enabled	1			Enable																										
D	RW	ENDRX				Enable or disable interrupt for ENDRX event																										
						See <a href="#">EVENTS_ENDRX</a>																										

### 35.10.3 INTENSET

Address offset: 0x304

## Enable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																L	J	I	H					G	F	E	D	C	B	A				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value																										Description					
A RW CTS																												Write '1' to Enable interrupt for CTS event						
																												See <a href="#">EVENTS_CTS</a>						
																												Set						
																												1						
																												Enable						
																												Disabled						
																												0						
																												Read: Disabled						
																												Enabled						
																												1						
B RW NCTS																												Write '1' to Enable interrupt for NCTS event						
																												See <a href="#">EVENTS_NCTS</a>						
																												Set						
																												1						
																												Enable						
																												Disabled						
																												0						
																												Read: Enabled						
C RW RXDRDY																											Write '1' to Enable interrupt for RXDRDY event							

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			L	J	I	H	G	F	E	D	C	B	A																					
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Id	RW	Field	Value	Id	Value	Description																												
						See <a href="#">EVENTS_RXRDY</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
D	RW	ENDRX				Write '1' to Enable interrupt for ENDRX event																												
						See <a href="#">EVENTS_ENDRX</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
E	RW	TXRDY				Write '1' to Enable interrupt for TXRDY event																												
						See <a href="#">EVENTS_TXRDY</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
F	RW	ENDTX				Write '1' to Enable interrupt for ENDTX event																												
						See <a href="#">EVENTS_ENDTX</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
G	RW	ERROR				Write '1' to Enable interrupt for ERROR event																												
						See <a href="#">EVENTS_ERROR</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
H	RW	RXTO				Write '1' to Enable interrupt for RXTO event																												
						See <a href="#">EVENTS_RXTO</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
I	RW	RXSTARTED				Write '1' to Enable interrupt for RXSTARTED event																												
						See <a href="#">EVENTS_RXSTARTED</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
J	RW	TXSTARTED				Write '1' to Enable interrupt for TXSTARTED event																												
						See <a href="#">EVENTS_TXSTARTED</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												
L	RW	TXSTOPPED				Write '1' to Enable interrupt for TXSTOPPED event																												
						See <a href="#">EVENTS_TXSTOPPED</a>																												
		Set	1			Enable																												
		Disabled	0			Read: Disabled																												
		Enabled	1			Read: Enabled																												

### 35.10.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				L	J	I	H	G	F	E	D	C	B	A																					
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																														
A	RW	CTS			Write '1' to Disable interrupt for CTS event																														
					See <a href="#">EVENTS_CTS</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
B	RW	NCTS			Write '1' to Disable interrupt for NCTS event																														
					See <a href="#">EVENTS_NCTS</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
C	RW	RXDRDY			Write '1' to Disable interrupt for RXDRDY event																														
					See <a href="#">EVENTS_RXDRDY</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
D	RW	ENDRX			Write '1' to Disable interrupt for ENDRX event																														
					See <a href="#">EVENTS_ENDRX</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
E	RW	TXDRDY			Write '1' to Disable interrupt for TXDRDY event																														
					See <a href="#">EVENTS_TXDRDY</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
F	RW	ENDTX			Write '1' to Disable interrupt for ENDTX event																														
					See <a href="#">EVENTS_ENDTX</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
G	RW	ERROR			Write '1' to Disable interrupt for ERROR event																														
					See <a href="#">EVENTS_ERROR</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
H	RW	RXTO			Write '1' to Disable interrupt for RXTO event																														
					See <a href="#">EVENTS_RXTO</a>																														
			Clear	1	Disable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
I	RW	RXSTARTED			Write '1' to																														

### **35.10.5 ERRORSRC**

Address offset: 0x480

### Error source

### **35.10.6 ENABLE**

Address offset: 0x500

## Enable UART

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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		Disabled				Enable or disable UARTE																								
				0				Disable UARTE																								
				8				Enable UARTE																								

### **35.10.7 PSEL.RTS**

Address offset: 0x508

Pin select for RTS signal

### **35.10.8 PSEL.TXD**

Address offset: 0x50C

## Pin select for TXD signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																										A	A	A	A	A	
Reset 0xFFFFFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value		Description																										
A	RW	PIN		[0..31]		Pin number																										
B	RW	CONNECT				Connection																										
			Disconnected	1		Disconnect																										
			Connected	0		Connect																										

## 35.10.9 PSEL.CTS

Address offset: 0x510

Pin select for CTS signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **35.10.10 PSEL.RXD**

Address offset: 0x514

Pin select for RXD signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **35.10.11 BAUDRATE**

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

35.10.12 RXD.PTR

Address offset: 0x534

Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PTR			Data pointer																											

### **35.10.13 RXD.MAXCNT**

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	MAXCNT			Maximum number of bytes in receive buffer

### **35.10.14 RXD.AMOUNT**

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	AMOUNT			Number of bytes transferred in the last transaction																											

35.10.15 TXD.PTR

Address offset: 0x544

Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PTR			Data pointer																											

### **35.10.16 TXD.MAXCNT**

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	MAXCNT			Maximum number of bytes in transmit buffer																											

**35.10.17 TXD.AMOUNT**

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	R	AMOUNT			Number of bytes transferred in the last transaction																											

## **35.10.18 CONFIG**

Address offset: 0x56C

## Configuration of parity and hardware flow control

### **35.11 Electrical specification**

### **35.11.1 UARTE electrical specification**

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{UARTE}}$	Baud rate for UARTE <sup>32</sup> .			1000	kbps
$I_{\text{UARTE1M}}$	Run current at max baud rate.		55		µA
$I_{\text{UARTE115k}}$	Run current at 115200 bps.		55		µA
$I_{\text{UARTE1k2}}$	Run current at 1200 bps.		55		µA
$I_{\text{UARTE,IDL}}$	Idle current for UARTE (STARTed, no XXX activity)		1		µA
$t_{\text{UARTE,CTSH}}$	CTS high time	1			µs

---

<sup>32</sup> Higher baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

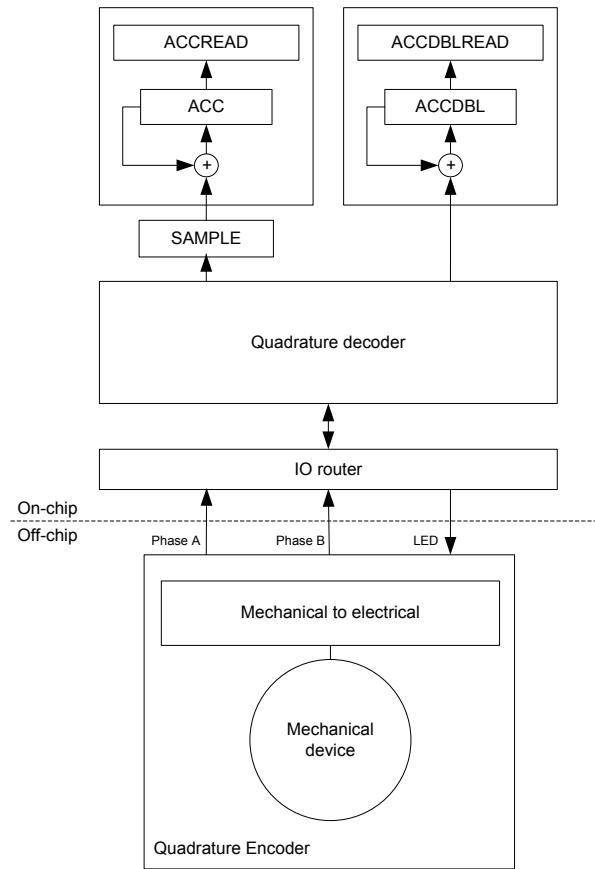
Symbol	Description	Min.	Typ.	Max.	Units
$t_{UARTE,START,LP}$	Time from STARTRX/STARTTX task to transmission started, low power mode		$t_{UARTE,STAR}$		$\mu s$
$t_{UARTE,START,CL}$	Time from STARTRX/STARTTX task to transmission started, constant latency mode		+	$t_{START\_HFIN}$	1 $\mu s$

## 36 QDEC — Quadrature decoder

The Quadrature decoder (QDEC) provides buffered decoding of quadrature-encoded sensor signals. It is suitable for mechanical and optical sensors.

The sample period and accumulation are configurable to match application requirements. The QDEC provides the following:

- Decoding of digital waveform from off-chip quadrature encoder.
- Sample accumulation eliminating hard real-time requirements to be enforced on application.
- Optional input de-bounce filters.
- Optional LED output signal for optical encoders.



**Figure 97: Quadrature decoder configuration**

### 36.1 Sampling and decoding

The QDEC decodes the output from an incremental motion encoder by sampling the QDEC phase input pins (A and B).

The off-chip quadrature encoder is an incremental motion encoder outputting two waveforms, phase A and phase B. The two output waveforms are always 90 degrees out of phase, meaning that one always changes level before the other. The direction of movement is indicated by which of these two waveforms that changes level first. Invalid transitions may occur, that is when the two waveforms switch simultaneously. This may occur if the wheel rotates too fast relative to the sample rate set for the decoder.

The QDEC decodes the output from the off-chip encoder by sampling the QDEC phase input pins (A and B) at a fixed rate as specified in the SAMPLEPER register.

If the SAMPLEPER value needs to be changed, the QDEC shall be stopped using the STOP task. SAMPLEPER can be then changed upon receiving the STOPPED event, and QDEC can be restarted using the START task. Failing to do so may result in unpredictable behaviour.

It is good practice to change other registers (LEDPOL, REPORTPER, DBFEN and LEDPRE) only when the QDEC is stopped.

When started, the decoder continuously samples the two input waveforms and decodes these by comparing the current sample pair (n) with the previous sample pair (n-1).

The decoding of the sample pairs is described in the table below.

**Table 83: Sampled value encoding**

Previous sample pair(n - 1)		Current samples pair(n)		SAMPLE register	ACC operation	ACCDBL operation	Description
A	B	A	B				
0	0	0	0	0	No change	No change	No movement
0	0	0	1	1	Increment	No change	Movement in positive direction
0	0	1	0	-1	Decrement	No change	Movement in negative direction
0	0	1	1	2	No change	Increment	Error: Double transition
0	1	0	0	-1	Decrement	No change	Movement in negative direction
0	1	0	1	0	No change	No change	No movement
0	1	1	0	2	No change	Increment	Error: Double transition
0	1	1	1	1	Increment	No change	Movement in positive direction
1	0	0	0	1	Increment	No change	Movement in positive direction
1	0	0	1	2	No change	Increment	Error: Double transition
1	0	1	0	0	No change	No change	No movement
1	0	1	1	-1	Decrement	No change	Movement in negative direction
1	1	0	0	2	No change	Increment	Error: Double transition
1	1	0	1	-1	Decrement	No change	Movement in negative direction
1	1	1	0	1	Increment	No change	Movement in positive direction
1	1	1	1	0	No change	No change	No movement

## 36.2 LED output

The LED output follows the sample period, and the LED is switched on a given period before sampling and switched off immediately after the inputs are sampled. The period the LED is switched on before sampling is given in the LEDPRE register.

The LED output pin polarity is specified in the LEDPOL register.

For using off-chip mechanical encoders not requiring a LED, the LED output can be disabled by writing value 'Disconnected' to the CONNECT field of the PSEL.LED register. In this case the QDEC will not acquire access to a LED output pin and the pin can be used for other purposes by the CPU.

## 36.3 Debounce filters

Each of the two-phase inputs have digital debounce filters.

When enabled through the DBFEN register, the filter inputs are sampled at a fixed 1 MHz frequency during the entire sample period (which is specified in the SAMPLEPER register), and the filters require all of the samples within this sample period to equal before the input signal is accepted and transferred to the output of the filter.

As a result, only input signal with a steady state longer than twice the period specified in SAMPLEPER are guaranteed to pass through the filter, and any signal with a steady state shorter than SAMPLEPER will always be suppressed by the filter. (This is assumed that the frequency during the debounce period never exceeds 500 kHz (as required by the Nyquist theorem when using a 1 MHz sample frequency)).

The LED will always be ON when the debounce filters are enabled, as the inputs in this case will be sampled continuously.

Note that when the debounce filters are enabled, displacements reported by the QDEC peripheral are delayed by one SAMPLEPER period.

## 36.4 Accumulators

The quadrature decoder contains two accumulator registers, ACC and ACCDBL, that accumulate respectively valid motion sample values and the number of detected invalid samples (double transitions).

The ACC register will accumulate all valid values (1/-1) written to the SAMPLE register. This can be useful for preventing hard real-time requirements from being enforced on the application. When using the ACC register the application does not need to read every single sample from the SAMPLE register, but can instead fetch the ACC register whenever it fits the application. The ACC register will always hold the relative movement of the external mechanical device since the previous clearing of the ACC register. Sample values indicating a double transition (2) will not be accumulated in the ACC register.

An ACCOF event will be generated if the ACC receives a SAMPLE value that would cause the register to overflow or underflow. Any SAMPLE value that would cause an ACC overflow or underflow will be discarded, but any samples not causing the ACC to overflow or underflow will still be accepted.

The accumulator ACCDBL accumulates the number of detected double transitions since the previous clearing of the ACCDBL register.

The ACC and ACCDBL registers can be cleared by the RDCLRACC and subsequently read using the ACCREAD and ACCDBLREAD registers.

The ACC register can be separately cleared by the RDCLRACC and subsequently read using the ACCREAD registers.

The ACCDBL register can be separately cleared by the RDCLRDBL and subsequently read using the ACCDBLREAD registers.

The REPORTPER register allows automating the capture of several samples before it can send out a REPORTRDY event in case a non-null displacement has been captured and accumulated, and a DBLRDY event in case one or more double-displacements have been captured and accumulated. The REPORTPER field in this register selects after how many samples the accumulators contents are evaluated to send (or not) REPORTRDY and DBLRDY events.

Using the RDCLRACC task (manually sent upon receiving the event, or using the DBLRDY\_RDCLRACC shortcut), ACCREAD can then be read.

In case at least one double transition has been captured and accumulated, a DBLRDY event is sent. Using the RDCLRDBL task (manually sent upon receiving the event, or using the DBLRDY\_RDCLRDBL shortcut), ACCDBLREAD can then be read.

## 36.5 Output/input pins

The QDEC uses a three-pin interface to the off-chip quadrature encoder.

These pins will be acquired when the QDEC is enabled in the ENABLE register. The pins acquired by the QDEC cannot be written by the CPU, but they can still be read by the CPU.

The pin numbers to be used for the QDEC are selected using the PSEL.n registers.

## 36.6 Pin configuration

The Phase A, Phase B, and LED signals are mapped to physical pins according to the configuration specified in the PSEL.A, PSEL.B, and PSEL.LED registers respectively.

If the CONNECT field value 'Disconnected' is specified in any of these registers, the associated signal will not be connected to any physical pin. The PSEL.A, PSEL.B, and PSEL.LED registers and their configurations are only used as long as the QDEC is enabled, and retained only as long as the device is in

ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN\_CNF[n] register.

To secure correct behavior in the QDEC, the pins used by the QDEC must be configured in the GPIO peripheral as described in [Table 84: GPIO configuration before enabling peripheral](#) on page 349 before enabling the QDEC. This configuration must be retained in the GPIO for the selected IOs as long as the QDEC is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

**Table 84: GPIO configuration before enabling peripheral**

QDEC signal	QDEC pin	Direction	Output value	Comment
Phase A	As specified in PSEL.A	Input	Not applicable	
Phase B	As specified in PSEL.B	Input	Not applicable	
LED	As specified in PSEL.LED	Input	Not applicable	

## 36.7 Registers

**Table 85: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40012000	QDEC	QDEC	Quadrature decoder	

**Table 86: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Task starting the quadrature decoder
TASKS_STOP	0x004	Task stopping the quadrature decoder
TASKS_READCLRACC	0x008	Read and clear ACC and ACCDBL
TASKS_RDCLRACC	0x00C	Read and clear ACC
TASKS_RDCLRD BL	0x010	Read and clear ACCDBL
EVENTS_SAMPLERDY	0x100	Event being generated for every new sample value written to the SAMPLE register
EVENTS_REPORTRDY	0x104	Non-null report ready
EVENTS_ACCOF	0x108	ACC or ACCDBL register overflow
EVENTS_DBLRDY	0x10C	Double displacement(s) detected
EVENTS_STOPPED	0x110	QDEC has been stopped
<b>SHORTS</b>	0x200	Shortcut register
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>ENABLE</b>	0x500	Enable the quadrature decoder
<b>LEDPOL</b>	0x504	LED output pin polarity
<b>SAMPLEPER</b>	0x508	Sample period
<b>SAMPLE</b>	0x50C	Motion sample value
<b>REPORTPER</b>	0x510	Number of samples to be taken before REPORTRDY and DBLRDY events can be generated
<b>ACC</b>	0x514	Register accumulating the valid transitions
<b>ACCREAD</b>	0x518	Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task
<b>PSEL.LED</b>	0x51C	Pin select for LED signal
<b>PSEL.A</b>	0x520	Pin select for A signal
<b>PSEL.B</b>	0x524	Pin select for B signal
<b>DBFEN</b>	0x528	Enable input debounce filters
<b>LEDPRE</b>	0x540	Time period the LED is switched ON prior to sampling
<b>ACCDBL</b>	0x544	Register accumulating the number of detected double transitions
<b>ACCDBLREAD</b>	0x548	Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRD BL task

### 36.7.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Id				G F E D C B A	
<b>Reset 0x00000000</b>				<b>0 0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A RW REPORTRDY_READCLRACC					Shortcut between REPORTRDY event and READCLRACC task
					See <a href="#">EVENTS_REPORTRDY</a> and <a href="#">TASKS_READCLRACC</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
B RW SAMPLERDY_STOP					Shortcut between SAMPLERDY event and STOP task
					See <a href="#">EVENTS_SAMPLERDY</a> and <a href="#">TASKS_STOP</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
C RW REPORTRDY_RDCLRACC					Shortcut between REPORTRDY event and RDCLRACC task
					See <a href="#">EVENTS_REPORTRDY</a> and <a href="#">TASKS_RDCLRACC</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
D RW REPORTRDY_STOP					Shortcut between REPORTRDY event and STOP task
					See <a href="#">EVENTS_REPORTRDY</a> and <a href="#">TASKS_STOP</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
E RW DBLRDY_RDCLRDBL					Shortcut between DBLRDY event and RDCLRDBL task
					See <a href="#">EVENTS_DBLRDY</a> and <a href="#">TASKS_RDCLRDBL</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
F RW DBLRDY_STOP					Shortcut between DBLRDY event and STOP task
					See <a href="#">EVENTS_DBLRDY</a> and <a href="#">TASKS_STOP</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut
G RW SAMPLERDY_READCLRACC					Shortcut between SAMPLERDY event and READCLRACC task
					See <a href="#">EVENTS_SAMPLERDY</a> and <a href="#">TASKS_READCLRACC</a>
				Disabled 0	Disable shortcut
				Enabled 1	Enable shortcut

### 36.7.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Id				E D C B A	
<b>Reset 0x00000000</b>				<b>0 0</b>	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A RW SAMPLERDY					Write '1' to Enable interrupt for SAMPLERDY event
					See <a href="#">EVENTS_SAMPLERDY</a>
				Set 1	Enable
				Disabled 0	Read: Disabled
				Enabled 1	Read: Enabled
B RW REPORTRDY					Write '1' to Enable interrupt for REPORTRDY event
					See <a href="#">EVENTS_REPORTRDY</a>
				Set 1	Enable
				Disabled 0	Read: Disabled
				Enabled 1	Read: Enabled
C RW ACCOF					Write '1' to Enable interrupt for ACCOF event
					See <a href="#">EVENTS_ACCOF</a>
				Set 1	Enable

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id		E D C B A				
Reset 0x00000000		0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
D	RW	DBLRDY				Write '1' to Enable interrupt for DBLRDY event
						See <a href="#">EVENTS_DBLRDY</a>
			Set		1	Enable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
E	RW	STOPPED				Write '1' to Enable interrupt for STOPPED event
						See <a href="#">EVENTS_STOPPED</a>
			Set		1	Enable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled

### 36.7.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id		E D C B A				
Reset 0x00000000		0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>
A	RW	SAMPLERDY				Write '1' to Disable interrupt for SAMPLERDY event
						See <a href="#">EVENTS_SAMPLERDY</a>
			Clear		1	Disable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
B	RW	REPORTRDY				Write '1' to Disable interrupt for REPORTRDY event
						See <a href="#">EVENTS_REPORTRDY</a>
			Clear		1	Disable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
C	RW	ACCOF				Write '1' to Disable interrupt for ACCOF event
						See <a href="#">EVENTS_ACCOF</a>
			Clear		1	Disable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
D	RW	DBLRDY				Write '1' to Disable interrupt for DBLRDY event
						See <a href="#">EVENTS_DBLRDY</a>
			Clear		1	Disable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled
E	RW	STOPPED				Write '1' to Disable interrupt for STOPPED event
						See <a href="#">EVENTS_STOPPED</a>
			Clear		1	Disable
			Disabled		0	Read: Disabled
			Enabled		1	Read: Enabled

### 36.7.4 ENABLE

Address offset: 0x500

Enable the quadrature decoder

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	ENABLE		Enable or disable the quadrature decoder																												
				When enabled the decoder pins will be active. When disabled the quadrature decoder pins are not active and can be used as GPIO .																												
			Disabled	0	Disable																											
			Enabled	1	Enable																											

### **36.7.5 LEDPOL**

Address offset: 0x504

## LED output pin polarity

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id	A																																
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																											
A	RW	LEDPOL				LED output pin polarity																											
		ActiveLow		0		Led active on output pin low																											
		ActiveHigh		1		Led active on output pin high																											

### **36.7.6 SAMPLEPER**

Address offset: 0x508

## Sample period

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Id																																					
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	0
Id	RW	Field	Value Id																												Description						
A	RW	SAMPLEPER	Sample period. The SAMPLE register will be updated for every new sample																																		
		128us	0																											128 us							
		256us	1																											256 us							
		512us	2																											512 us							
		1024us	3																											1024 us							
		2048us	4																											2048 us							
		4096us	5																											4096 us							
		8192us	6																											8192 us							
		16384us	7																											16384 us							
		32ms	8																											32768 us							
		65ms	9																											65536 us							
		131ms	10																											131072 us							

### **36.7.7 SAMPLE**

Address offset: 0x50C

### Motion sample value

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	SAMPLE		[-1..2]	Last motion sample																											

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	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	
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	
Id	RW	Field	Value Id	Value	Description																											
					The value is a 2's complement value, and the sign gives the direction of the motion. The value '2' indicates a double transition.																											

## **36.7.8 REPORTPER**

Address offset: 0x510

Number of samples to be taken before REPORTRDY and DBLRDY events can be generated

### 36.7.9 ACC

Address offset: 0x514

Register accumulating the valid transitions

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	0	0
Id	RW	Field	Value	Id	Value	Description																												
A	R	ACC			[-1024..1023]	Register accumulating all valid samples (not double transition) read from the SAMPLE register																												
						Double transitions ( SAMPLE = 2 ) will not be accumulated in this register. The value is a 32 bit 2's complement value. If a sample that would cause this register to overflow or underflow is received, the sample will be ignored and an overflow event ( ACCOF ) will be generated. The ACC register is cleared by triggering the READCLRACC or the RDCLRACC task.																												

### **36.7.10 ACCREAD**

Address offset: 0x518

Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	R	ACCREAD		[-1024..1023]	Snapshot of the ACC register.																											
					The ACCREAD register is updated when the READCLRACC or RDCLRACC task is triggered																											

### **36.7.11 PSEL.LED**

Address offset: 0x51C

Pin select for LED signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																										A	A	A	A	A	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **36.7.12 PSEL.A**

Address offset: 0x520

Pin select for A signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																										A	A	A	A	A	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

## **36.7.13 PSEL.B**

Address offset: 0x524

## Pin select for B signal

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

### **36.7.14 DBFEN**

Address offset: 0x528

## Enable input debounce filters

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	DBFEN			Enable input debounce filters
		Disabled		0	Debounce input filters disabled
		Enabled		1	Debounce input filters enabled

### 36.7.15 LEDPRE

Address offset: 0x540

Time period the LED is switched ON prior to sampling

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A A A A				
Reset 0x00000010	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	LEDPRE		[1..511]	Period in us the LED is switched on prior to sampling

### 36.7.16 ACCDBL

Address offset: 0x544

Register accumulating the number of detected double transitions

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	R	ACCDBL		[0..15]	Register accumulating the number of detected double or illegal transitions. ( SAMPLE = 2 ).
					When this register has reached its maximum value the accumulation of double / illegal transitions will stop. An overflow event ( ACCOF ) will be generated if any double or illegal transitions are detected after the maximum value was reached. This field is cleared by triggering the READCLRACC or RDCLRDBL task.

### 36.7.17 ACCDBLREAD

Address offset: 0x548

Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	R	ACCDBLREAD		[0..15]	Snapshot of the ACCDBL register. This field is updated when the READCLRACC or RDCLRDBL task is triggered.

## 36.8 Electrical specification

### 36.8.1 QDEC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>QDEC</sub>	Run current		5		µA
t <sub>SAMPLE</sub>	Time between sampling signals from quadrature decoder		128	131072	µs
t <sub>LED</sub>	Time from LED is turned on to signals are sampled	0	511		µs

# 37 SAADC — Successive approximation analog-to-digital converter

The ADC is a differential successive approximation register (SAR) analog-to-digital converter.

Listed here are the main features of SAADC:

- 8/10/12-bit resolution, 14-bit resolution with oversampling
- Up to eight input channels
  - One channel per single-ended input and two channels per differential input
  - Scan mode can be configured with both single-ended channels and differential channels.
- Full scale input range (0 to VDD)
- Sampling triggered via a task from software or a PPI channel for full flexibility on sample frequency source from low power 32.768kHz RTC or more accurate 1/16MHz Timers
- One-shot conversion mode to sample a single channel
- Scan mode to sample a series of channels in sequence. Sample delay between channels is  $t_{ack} + t_{conv}$  which may vary between channels according to user configuration of  $t_{ack}$ .
- Support for direct sample transfer to RAM using EasyDMA
- Interrupts on single sample and full buffer events
- Samples stored as 16-bit 2's complement values for differential and single-ended sampling
- Continuous sampling without the need of an external timer
- Internal resistor string
- Limit checking on the fly

## 37.1 Shared resources

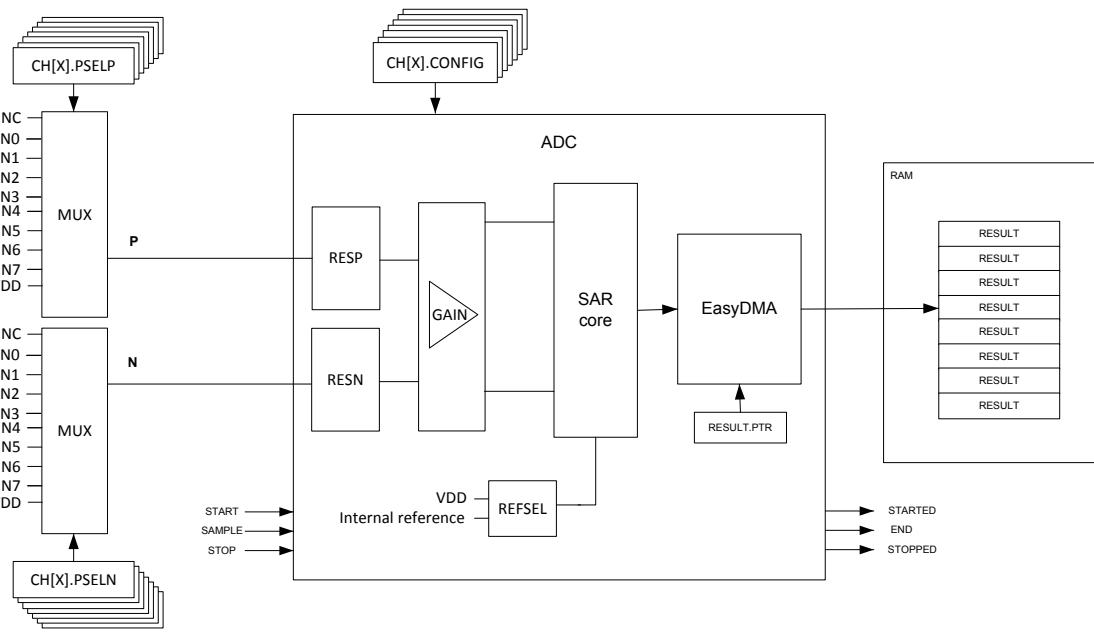
The ADC can coexist with COMP, LPCOMP and other peripherals using one of AIN0–AIN7, provided these are assigned to different pins.

It is not recommended to select the same analog input pin for both modules.

## 37.2 Overview

The ADC supports up to eight external analog input channels, depending on package variant. It can be operated in a one-shot mode with sampling under software control, or a continuous conversion mode with a programmable sampling rate.

The analog inputs can be configured as eight single-ended inputs, four differential inputs or a combination of these. Each channel can be configured to select AIN0 to AIN7 pins, or the VDD pin. Channels can be sampled individually in one-shot or continuous sampling modes, or, using scan mode, multiple channels can be sampled in sequence. Channels can also be oversampled to improve noise performance.



**Figure 98: Simplified ADC block diagram**

Internally, the ADC is always a differential analog-to-digital converter, but by default it is configured with single-ended input in the MODE field of the CH[n].CONFIG register. In single-ended mode, the negative input will be shorted to ground internally.

The assumption in single-ended mode is that the internal ground of the ADC is the same as the external ground that the measured voltage is referred to. The ADC is thus sensitive to ground bounce on the PCB in single-ended mode. If this is a concern we recommend using differential measurement.

### 37.3 Digital output

The output result of the ADC depends on the settings in the CH[n].CONFIG and RESOLUTION registers as follows:

$$\text{RESULT} = [V(P) - V(N)] * \text{GAIN/REFERENCE} * 2^{(\text{RESOLUTION} - m)}$$

where

**V(P)**

is the voltage at input P

**V(N)**

is the voltage at input N

**GAIN**

is the selected gain setting

**REFERENCE**

is the selected reference voltage

and  $m=0$  if CONFIG.MODE=SE, or  $m=1$  if CONFIG.MODE=Diff.

The result generated by the ADC will deviate from the expected due DC errors like offset, gain, differential non-linearity (DNL), and integral non-linearity (INL). See [Electrical specification](#) for details on these parameters. The result can also vary due to AC errors like non-linearities in the GAIN block, settling errors due to high source impedance and sampling jitter. For battery measurement the DC errors are most noticeable.

The ADC has a wide selection of gains controlled in the GAIN field of the CH[n].CONFIG register. If CH[n].CONFIG.REFSEL=0, the input range of the ADC core is nominally  $\pm 0.6$  V differential and the input must be scaled accordingly.

The ADC has a temperature dependent offset. If the ADC is to operate over a large temperature range, we recommend running CALIBRATEOFFSET at regular intervals, a CALIBRATEDONE event will be fired when the calibration is complete

## 37.4 Analog inputs and channels

Up to eight analog input channels, CH[n](n=0..7), can be configured.

See [Shared resources](#) on page 356 for shared input with comparators.

Any one of the available channels can be enabled for the ADC to operate in one-shot mode. If more than one CH[n] is configured, the ADC enters scan mode.

An analog input is selected as a positive converter input if CH[n].PSELP is set, setting CH[n].PSELP also enables the particular channel.

An analog input is selected as a negative converter input if CH[n].PSELN is set. The CH[n].PSELN register will have no effect unless differential mode is enabled, see MODE field in CH[n].CONFIG register.

If more than one of the CH[n].PSELP registers is set, the device enters scan mode. Input selections in scan mode are controlled by the CH[n].PSELP and CH[n].PSELN registers, where CH[n].PSELN is only used if the particular scan channel is specified as differential, see MODE field in CH[n].CONFIG register.

**Important:** Channels selected for either COMP or LPCOMP cannot be used at the same time for ADC sampling, though channels not selected for use by these blocks can be used by the ADC.

**Table 87: Legal connectivity CH[n] vs. analog input**

Channel input	Source	Connectivity
CH[n].PSELP	AIN0...AIN7	Yes(any)
CH[n].PSELP	VDD	Yes
CH[n].PSELN	AIN0...AIN7	Yes(any)
CH[n].PSELN	VDD	Yes

## 37.5 Operation modes

The ADC input configuration supports one-shot mode, continuous mode and scan mode.

Scan mode and oversampling cannot be combined.

### 37.5.1 One-shot mode

One-shot operation is configured by enabling only one of the available channels defined by CH[n].PSELP, CH[n].PSELN, and CH[n].CONFIG registers.

Upon a SAMPLE task, the ADC starts to sample the input voltage. The CH[n].CONFIG.TACQ controls the acquisition time.

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual value has been transferred into RAM by EasyDMA. For more information, see [EasyDMA](#) on page 360.

### 37.5.2 Continuous mode

Continuous sampling can be achieved by using the internal timer in the ADC, or triggering the SAMPLE task from one of the general purpose timers through the PPI.

Care shall be taken to ensure that the sample rate fulfills the following criteria, depending on how many channels are active:

$$f_{\text{SAMPLE}} < 1 / [\tau_{\text{ACQ}} + \tau_{\text{conv}}]$$

The SAMPLERATE register can be used as a local timer instead of triggering individual SAMPLE tasks. When SAMPLERATE.MODE is set to Timers, it is sufficient to trigger SAMPLE task only once in order to start the SAADC and triggering the STOP task will stop sampling. The SAMPLERATE.CC field controls the sample rate.

The SAMPLERATE timer mode cannot be combined with SCAN mode, and only one channel can be enabled in this mode.

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual value has been transferred into RAM by EasyDMA.

### 37.5.3 Oversampling

An accumulator in the ADC can be used to average noise on the analog input. In general, oversampling improves the signal-to-noise ratio (SNR). Oversampling, however, does not improve the integral non-linearity (INL), or differential non-linearity (DNL).

Oversampling and scan should not be combined, since oversampling and scan will average over input channels.

The accumulator is controlled in the OVERSAMPLE register. The SAMPLE task must be set  $2^{\text{OVERSAMPLE}}$  number of times before the result is written to RAM. This can be achieved by:

- Configuring a fixed sampling rate using the local timer or a general purpose timer and PPI to trigger a SAMPLE task
- Triggering SAMPLE  $2^{\text{OVERSAMPLE}}$  times from software
- Enabling BURST mode

CH[n].CONFIG.BURST can be enabled to avoid setting SAMPLE task  $2^{\text{OVERSAMPLE}}$  times. With BURST = 1 the ADC will sample the input  $2^{\text{OVERSAMPLE}}$  times as fast as it can (actual timing:  $<(t_{\text{ACQ}}+t_{\text{CONV}}) \times 2^{\text{OVERSAMPLE}}$ ). Thus, for the user it will just appear like the conversion took a bit longer time, but other than that, it is similar to one-shot mode. Scan mode can be combined with BURST=1, if burst is enabled on all channels.

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event signals that enough conversions have taken place for an oversampled result to get transferred into RAM. Note that both events may occur before the actual value has been transferred into RAM by EasyDMA.

### 37.5.4 Scan mode

A channel is considered enabled if CH[n].PSELP is set. If more than one channel, CH[n], is enabled, the ADC enters scan mode.

In scan mode, one SAMPLE task will trigger one conversion per enabled channel. The time it takes to sample all channels is:

```
Total time < Sum(CH[x].tACQ+tCONV), x=0..enabled channels
```

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event signals has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual values have been transferred into RAM by EasyDMA.

*Figure 99: Example of RAM placement (even RESULT.MAXCNT), channels 1, 2 and 5 enabled* on page 360 provides an example of results placement in Data RAM, with an even RESULT.MAXCNT. In this example, channels 1, 2 and 5 are enabled, all others are disabled.

	31	16 15	0
RESULT.PTR	CH[2] 1 <sup>st</sup> result	CH[1] 1 <sup>st</sup> result	
RESULT.PTR + 4	CH[1] 2 <sup>nd</sup> result	CH[5] 1 <sup>st</sup> result	
RESULT.PTR + 8	CH[5] 2 <sup>nd</sup> result	CH[2] 2 <sup>nd</sup> result	
(...)			
RESULT.PTR + 2*(RESULT.MAXCNT - 2)	CH[5] last result	CH[2] last result	

**Figure 99: Example of RAM placement (even RESULT.MAXCNT), channels 1, 2 and 5 enabled**

*Figure 100: Example of RAM placement (odd RESULT.MAXCNT), channels 1, 2 and 5 enabled* on page 360 provides an example of results placement in Data RAM, with an odd RESULT.MAXCNT. In this example, channels 1, 2 and 5 are enabled, all others are disabled. The last 32-bit word is populated only with one 16-bit result.

	31	16 15	0
RESULT.PTR	CH[2] 1 <sup>st</sup> result	CH[1] 1 <sup>st</sup> result	
RESULT.PTR + 4	CH[1] 2 <sup>nd</sup> result	CH[5] 1 <sup>st</sup> result	
RESULT.PTR + 8	CH[5] 2 <sup>nd</sup> result	CH[2] 2 <sup>nd</sup> result	
(...)			
RESULT.PTR + 2*(RESULT.MAXCNT - 1)		CH[5] last result	

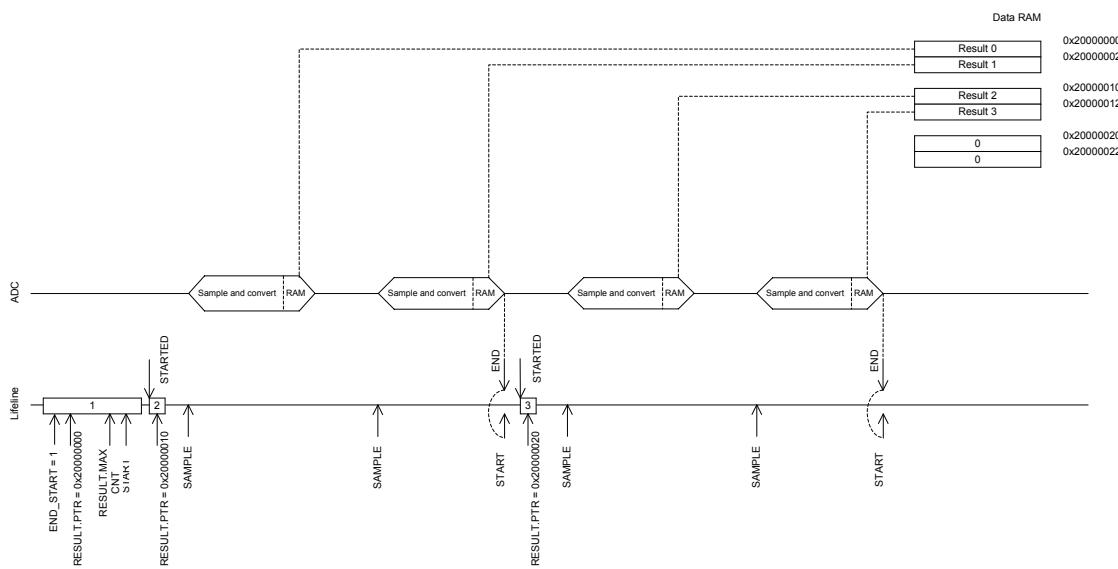
**Figure 100: Example of RAM placement (odd RESULT.MAXCNT), channels 1, 2 and 5 enabled**

## 37.6 EasyDMA

After configuring RESULT.PTR and RESULT.MAXCNT, the ADC resources are started by triggering the START task. The ADC is using EasyDMA to store results in a Result buffer in RAM.

The Result buffer is located at the address specified in the RESULT.PTR register. The RESULT.PTR register is double-buffered and it can be updated and prepared for the next START task immediately after the STARTED event is generated. The size of the Result buffer is specified in the RESULT.MAXCNT register and the ADC will generate an END event when it has filled up the Result buffer, see *Figure 101: ADC* on page 361. Results are stored in little-endian byte order in Data RAM. Every sample will be sign extended to 16 bit before stored in the Result buffer.

The ADC is stopped by triggering the STOP task. The STOP task will terminate an ongoing sampling. The ADC will generate a STOPPED event when it has stopped. If the ADC is already stopped when the STOP task is triggered, the STOPPED event will still be generated.



**Figure 101: ADC**

If the RESULT.PTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

The EasyDMA will have finished accessing the RAM when the END or STOPPED event has been generated.

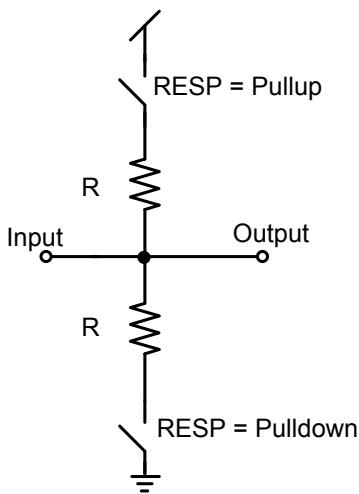
The RESULT.AMOUNT register can be read following an END event or a STOPPED event to see how many results have been transferred to the Result buffer in RAM since the START task was triggered.

In Scan mode, the size of the Result buffer must be large enough to have room for a minimum one result from each of the enabled channels. To secure this, RESULT.MAXCNT must be specified to  $\text{RESULT.MAXCNT} \geq \text{"number of channels enabled"}$ . See [Scan mode](#) on page 359 for more information about Scan mode.

## 37.7 Resistor ladder

The ADC has an internal resistor string for positive and negative input.

See [Figure 102: Resistor ladder for positive input \(negative input is equivalent, using RESN instead of RESP\)](#) on page 362. The resistors are controlled in the CH[n].CONFIG.RESP and CH[n].CONFIG.RESN registers.



**Figure 102: Resistor ladder for positive input (negative input is equivalent, using RESN instead of RESP)**

## 37.8 Reference

The ADC can use two different references, controlled in the REFSEL field of the CH[n].CONFIG register.

These are:

- Internal reference
- VDD as reference

The internal reference results in an input range of  $\pm 0.6$  V on the ADC core. VDD as reference results in an input range of  $\pm VDD/4$  on the ADC core. The gain block can be used to change the effective input range of the ADC.

$$\text{Input range} = (+- 0.6 \text{ V or } \pm VDD/4) / \text{Gain}$$

For example, choosing VDD as reference, single ended input (grounded negative input), and a gain of 1/4 the input range will be:

$$\text{Input range} = (VDD/4) / (1/4) = VDD$$

With internal reference, single ended input (grounded negative input), and a gain of 1/6 the input range will be:

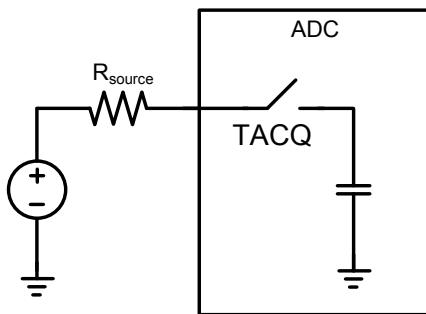
$$\text{Input range} = (0.6 \text{ V}) / (1/6) = 3.6 \text{ V}$$

The AIN0-AIN7 inputs cannot exceed VDD, or be lower than VSS.

## 37.9 Acquisition time

To sample the input voltage, the ADC connects a capacitor to the input.

For illustration, see [Figure 103: Simplified ADC sample network](#) on page 363. The acquisition time indicates how long the capacitor is connected, see TACQ field in CH[n].CONFIG register. The required acquisition time depends on the source ( $R_{source}$ ) resistance. For high source resistance the acquisition time should be increased, see [Table 88: Acquisition time](#) on page 363.



**Figure 103: Simplified ADC sample network**

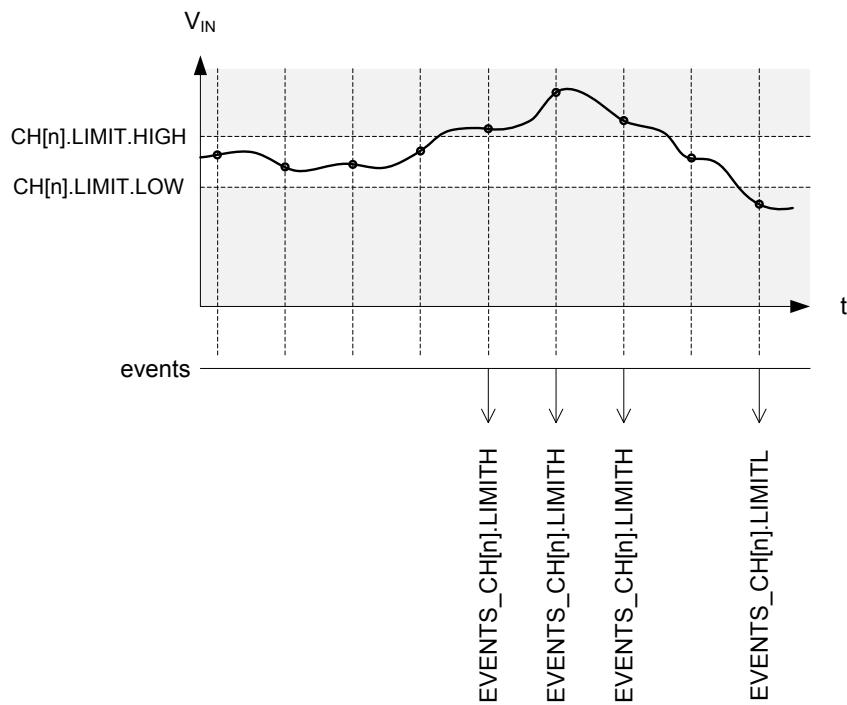
**Table 88: Acquisition time**

TACQ [μs]	Maximum source resistance [kOhm]
3	10
5	40
10	100
15	200
20	400
40	800

### 37.10 Limits event monitoring

A channel can be event monitored by configuring limit register CH[n].LIMIT.

If the conversion result is higher than the defined high limit, or lower than the defined low limit, the appropriate event will get fired.



**Figure 104: Example of limits monitoring on channel 'n'**

Note that when setting the limits, CH[n].LIMIT.HIGH shall always be higher than or equal to CH[n].LIMIT.LOW . In other words, an event can be fired only when the input signal has been sampled

outside of the defined limits. It is not possible to fire an event when the input signal is inside a defined range by swapping high and low limits.

The comparison to limits always takes place, there is no need to enable it. If comparison is not required on a channel, the software shall simply ignore the related events. In that situation, the value of the limits registers is irrelevant, so it does not matter if CH[n].LIMIT.LOW is lower than CH[n].LIMIT.HIGH or not.

## 37.11 Registers

**Table 89: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40007000	SAADC	SAADC	Analog to digital converter	

**Table 90: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start the ADC and prepare the result buffer in RAM
TASKS_SAMPLE	0x004	Take one ADC sample, if scan is enabled all channels are sampled
TASKS_STOP	0x008	Stop the ADC and terminate any on-going conversion
TASKS_CALIBRATEOFFSET	0x00C	Starts offset auto-calibration
EVENTS_STARTED	0x100	The ADC has started
EVENTS_END	0x104	The ADC has filled up the Result buffer
EVENTS_DONE	0x108	A conversion task has been completed. Depending on the mode, multiple conversions might be needed for a result to be transferred to RAM.
EVENTS_RESULTDONE	0x10C	A result is ready to get transferred to RAM.
EVENTS_CALIBRATEDON	0x110	Calibration is complete
EVENTS_STOPPED	0x114	The ADC has stopped
EVENTS_CH[0].LIMITH	0x118	Last results is equal or above CH[0].LIMIT.HIGH
EVENTS_CH[0].LIMITL	0x11C	Last results is equal or below CH[0].LIMIT.LOW
EVENTS_CH[1].LIMITH	0x120	Last results is equal or above CH[1].LIMIT.HIGH
EVENTS_CH[1].LIMITL	0x124	Last results is equal or below CH[1].LIMIT.LOW
EVENTS_CH[2].LIMITH	0x128	Last results is equal or above CH[2].LIMIT.HIGH
EVENTS_CH[2].LIMITL	0x12C	Last results is equal or below CH[2].LIMIT.LOW
EVENTS_CH[3].LIMITH	0x130	Last results is equal or above CH[3].LIMIT.HIGH
EVENTS_CH[3].LIMITL	0x134	Last results is equal or below CH[3].LIMIT.LOW
EVENTS_CH[4].LIMITH	0x138	Last results is equal or above CH[4].LIMIT.HIGH
EVENTS_CH[4].LIMITL	0x13C	Last results is equal or below CH[4].LIMIT.LOW
EVENTS_CH[5].LIMITH	0x140	Last results is equal or above CH[5].LIMIT.HIGH
EVENTS_CH[5].LIMITL	0x144	Last results is equal or below CH[5].LIMIT.LOW
EVENTS_CH[6].LIMITH	0x148	Last results is equal or above CH[6].LIMIT.HIGH
EVENTS_CH[6].LIMITL	0x14C	Last results is equal or below CH[6].LIMIT.LOW
EVENTS_CH[7].LIMITH	0x150	Last results is equal or above CH[7].LIMIT.HIGH
EVENTS_CH[7].LIMITL	0x154	Last results is equal or below CH[7].LIMIT.LOW
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
STATUS	0x400	Status
ENABLE	0x500	Enable or disable ADC
CH[0].PSEL_P	0x510	Input positive pin selection for CH[0]
CH[0].PSEL_N	0x514	Input negative pin selection for CH[0]
CH[0].CONFIG	0x518	Input configuration for CH[0]
CH[0].LIMIT	0x51C	High/low limits for event monitoring a channel
CH[1].PSEL_P	0x520	Input positive pin selection for CH[1]
CH[1].PSEL_N	0x524	Input negative pin selection for CH[1]
CH[1].CONFIG	0x528	Input configuration for CH[1]
CH[1].LIMIT	0x52C	High/low limits for event monitoring a channel
CH[2].PSEL_P	0x530	Input positive pin selection for CH[2]

Register	Offset	Description
<i>CH[2].PSELN</i>	0x534	Input negative pin selection for CH[2]
<i>CH[2].CONFIG</i>	0x538	Input configuration for CH[2]
<i>CH[2].LIMIT</i>	0x53C	High/low limits for event monitoring a channel
<i>CH[3].PSELP</i>	0x540	Input positive pin selection for CH[3]
<i>CH[3].PSELN</i>	0x544	Input negative pin selection for CH[3]
<i>CH[3].CONFIG</i>	0x548	Input configuration for CH[3]
<i>CH[3].LIMIT</i>	0x54C	High/low limits for event monitoring a channel
<i>CH[4].PSELP</i>	0x550	Input positive pin selection for CH[4]
<i>CH[4].PSELN</i>	0x554	Input negative pin selection for CH[4]
<i>CH[4].CONFIG</i>	0x558	Input configuration for CH[4]
<i>CH[4].LIMIT</i>	0x55C	High/low limits for event monitoring a channel
<i>CH[5].PSELP</i>	0x560	Input positive pin selection for CH[5]
<i>CH[5].PSELN</i>	0x564	Input negative pin selection for CH[5]
<i>CH[5].CONFIG</i>	0x568	Input configuration for CH[5]
<i>CH[5].LIMIT</i>	0x56C	High/low limits for event monitoring a channel
<i>CH[6].PSELP</i>	0x570	Input positive pin selection for CH[6]
<i>CH[6].PSELN</i>	0x574	Input negative pin selection for CH[6]
<i>CH[6].CONFIG</i>	0x578	Input configuration for CH[6]
<i>CH[6].LIMIT</i>	0x57C	High/low limits for event monitoring a channel
<i>CH[7].PSELP</i>	0x580	Input positive pin selection for CH[7]
<i>CH[7].PSELN</i>	0x584	Input negative pin selection for CH[7]
<i>CH[7].CONFIG</i>	0x588	Input configuration for CH[7]
<i>CH[7].LIMIT</i>	0x58C	High/low limits for event monitoring a channel
<i>RESOLUTION</i>	0x5F0	Resolution configuration
<i>OVERSAMPLE</i>	0x5F4	Oversampling configuration. OVERSAMPLE should not be combined with SCAN. The RESOLUTION is applied before averaging, thus for high OVERSAMPLE a higher RESOLUTION should be used.
<i>SAMPLERATE</i>	0x5F8	Controls normal or continuous sample rate
<i>RESULT.PTR</i>	0x62C	Data pointer
<i>RESULT.MAXCNT</i>	0x630	Maximum number of buffer words to transfer
<i>RESULT.AMOUNT</i>	0x634	Number of buffer words transferred since last START

### 37.11.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A										
<b>Reset 0x00000000</b>		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Id	RW	Field	Value	Id	Value	Description																											
A RW STARTED						Enable or disable interrupt for STARTED event																											
						See <a href="#">EVENTS_STARTED</a>																											
				Disabled	0	Disable																											
				Enabled	1	Enable																											
B RW END						Enable or disable interrupt for END event																											
						See <a href="#">EVENTS_END</a>																											
				Disabled	0	Disable																											
				Enabled	1	Enable																											
C RW DONE						Enable or disable interrupt for DONE event																											
						See <a href="#">EVENTS_DONE</a>																											
				Disabled	0	Disable																											
				Enabled	1	Enable																											
D RW RESULTDONE						Enable or disable interrupt for RESULTDONE event																											
						See <a href="#">EVENTS_RESULTDONE</a>																											
				Disabled	0	Disable																											
				Enabled	1	Enable																											

				Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Id	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A										
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
<b>E</b>	RW	CALIBRATEDONE			Enable or disable interrupt for CALIBRATEDONE event																								See <a href="#">EVENTS_CALIBRATEDONE</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>F</b>	RW	STOPPED			Enable or disable interrupt for STOPPED event																								See <a href="#">EVENTS_STOPPED</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>G</b>	RW	CH0LIMITH			Enable or disable interrupt for CH[0].LIMITH event																								See <a href="#">EVENTS_CH[0].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>H</b>	RW	CH0LIMITL			Enable or disable interrupt for CH[0].LIMITL event																								See <a href="#">EVENTS_CH[0].LIMITL</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>I</b>	RW	CH1LIMITH			Enable or disable interrupt for CH[1].LIMITH event																								See <a href="#">EVENTS_CH[1].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>J</b>	RW	CH1LIMITL			Enable or disable interrupt for CH[1].LIMITL event																								See <a href="#">EVENTS_CH[1].LIMITL</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>K</b>	RW	CH2LIMITH			Enable or disable interrupt for CH[2].LIMITH event																								See <a href="#">EVENTS_CH[2].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>L</b>	RW	CH2LIMITL			Enable or disable interrupt for CH[2].LIMITL event																								See <a href="#">EVENTS_CH[2].LIMITL</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>M</b>	RW	CH3LIMITH			Enable or disable interrupt for CH[3].LIMITH event																								See <a href="#">EVENTS_CH[3].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>N</b>	RW	CH3LIMITL			Enable or disable interrupt for CH[3].LIMITL event																								See <a href="#">EVENTS_CH[3].LIMITL</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>O</b>	RW	CH4LIMITH			Enable or disable interrupt for CH[4].LIMITH event																								See <a href="#">EVENTS_CH[4].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>P</b>	RW	CH4LIMITL			Enable or disable interrupt for CH[4].LIMITL event																								See <a href="#">EVENTS_CH[4].LIMITL</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					
<b>Q</b>	RW	CH5LIMITH			Enable or disable interrupt for CH[5].LIMITH event																								See <a href="#">EVENTS_CH[5].LIMITH</a>							
					Disabled	0																									Disable					
					Enabled	1																									Enable					

## 37.11.2 INTENSET

Address offset: 0x304

## Enable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A										
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id																															
A RW STARTED			Value																										Description					
A RW STARTED			Write '1' to Enable interrupt for STARTED event																															
			See <a href="#">EVENTS_STARTED</a>																															
			Set																											Enable				
			Disabled																											Read: Disabled				
B RW END			Enabled																										Read: Enabled					
			Write '1' to Enable interrupt for END event																															
			See <a href="#">EVENTS_END</a>																										Set					
			Set																										Enable					
C RW DONE			Disabled																										Read: Disabled					
			Enabled																										Read: Enabled					
			Write '1' to Enable interrupt for DONE event																															
			See <a href="#">EVENTS_DONE</a>																										Set					
D RW RESULTDONE			Set																										Enable					
			Disabled																										Read: Disabled					
			Enabled																										Read: Enabled					
			See <a href="#">EVENTS_RESULTDONE</a>																											Set				



Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Id				V U T S R Q P O N M L K J I H G F E D C B A	
Reset 0x00000000				0 0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
					See <a href="#">EVENTS_CH[4].LIMITH</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to Enable interrupt for CH[4].LIMITL event
					See <a href="#">EVENTS_CH[4].LIMITL</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to Enable interrupt for CH[5].LIMITH event
					See <a href="#">EVENTS_CH[5].LIMITH</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to Enable interrupt for CH[5].LIMITL event
					See <a href="#">EVENTS_CH[5].LIMITL</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to Enable interrupt for CH[6].LIMITH event
					See <a href="#">EVENTS_CH[6].LIMITH</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to Enable interrupt for CH[6].LIMITL event
					See <a href="#">EVENTS_CH[6].LIMITL</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
U	RW	CH7LIMITH			Write '1' to Enable interrupt for CH[7].LIMITH event
					See <a href="#">EVENTS_CH[7].LIMITH</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to Enable interrupt for CH[7].LIMITL event
					See <a href="#">EVENTS_CH[7].LIMITL</a>
		Set	1	1	Enable
		Disabled	0	0	Read: Disabled
		Enabled	1	1	Read: Enabled

### 37.11.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Id				V U T S R Q P O N M L K J I H G F E D C B A	
Reset 0x00000000				0 0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	STARTED			Write '1' to Disable interrupt for STARTED event
					See <a href="#">EVENTS_STARTED</a>
		Clear	1	1	Disable

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Id			V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A									
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
<b>Id</b>			<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																								
									Disabled																								
									Enabled																								
<b>B</b>			<b>RW</b>	<b>END</b>					Write '1' to Disable interrupt for END event																								
									See <a href="#">EVENTS-END</a>																								
									Clear																								
									Disable																								
									Disabled																								
									Enabled																								
<b>C</b>			<b>RW</b>	<b>DONE</b>					Write '1' to Disable interrupt for DONE event																								
									See <a href="#">EVENTS-DONE</a>																								
									Clear																								
									Disable																								
									Disabled																								
									Enabled																								
<b>D</b>			<b>RW</b>	<b>RESULTDONE</b>					Write '1' to Disable interrupt for RESULTDONE event																								
									See <a href="#">EVENTS-RESULTDONE</a>																								
									Clear																								
									Disable																								
									Disabled																								
									Read: Enabled																								
<b>E</b>			<b>RW</b>	<b>CALIBRATEDONE</b>					Write '1' to Disable interrupt for CALIBRATEDONE event																								
									See <a href="#">EVENTS-CALIBRATEDONE</a>																								
									Clear																								
									Disable																								
									Disabled																								
									Read: Enabled																								
<b>F</b>			<b>RW</b>	<b>STOPPED</b>					Write '1' to Disable interrupt for STOPPED event																								
									See <a href="#">EVENTS-STOPPED</a>																								
									Clear																								
									Disable																								
									Disabled																								
									Read: Enabled																								
<b>G</b>			<b>RW</b>	<b>CH0LIMITH</b>					Write '1' to Disable interrupt for CH[0].LIMITH event																								
									See <a href="#">EVENTS-CH[0].LIMITH</a>																								
									Clear																								
									Disable																								
									Enabled																								
<b>H</b>			<b>RW</b>	<b>CH0LIMITL</b>					Write '1' to Disable interrupt for CH[0].LIMITL event																								
									See <a href="#">EVENTS-CH[0].LIMITL</a>																								
									Clear																								
									Disable																								
									Enabled																								
<b>I</b>			<b>RW</b>	<b>CH1LIMITH</b>					Write '1' to Disable interrupt for CH[1].LIMITH event																								
									See <a href="#">EVENTS-CH[1].LIMITH</a>																								



Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Id															V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value	Description																																
V	RW	CH7LIMITL			Write '1' to Disable interrupt for CH[7].LIMITL event																																
					See <a href="#">EVENTS_CH[7].LIMITL</a>																																
		Clear	1		Disable																																
		Disabled	0		Read: Disabled																																
		Enabled	1		Read: Enabled																																

### **37.11.4 STATUS**

Address offset: 0x400

## Status

### **37.11.5 ENABLE**

Address offset: 0x500

## Enable or disable ADC

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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 ADC																										
			Disabled	0		Disable ADC																										
			Enabled	1		Enable ADC																										
				When enabled, the ADC will acquire access to the analog input pins specified in the CH[n].PSEL and CH[n].PSELN registers.																												

37.11.6 CH[0].PSEL.P

Address offset: 0x510

Input positive pin selection for CH[0]

### **37.11.7 CH[0].PSELN**

Address offset: 0x514

Input negative pin selection for CH[0]

### **37.11.8 CH[0].CONFIG**

Address offset: 0x518

## Input configuration for CH[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														G	F	E	E	E	D	C	C	C	B	B	A	A								
Reset 0x00020000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value																											Description				
F RW MODE																												40us		5	40 us			
F RW MODE																												SE		0	Enable differential mode Single ended, PSELN will be ignored, negative input to ADC shorted to GND			
F RW MODE																												Diff		1	Differential			
G RW BURST																												Enable burst mode						
G RW BURST																												Disabled		0	Burst mode is disabled (normal operation)			
G RW BURST																												Enabled		1	Burst mode is enabled. SAADC takes $2^{\text{OVERSAMPLE}}$ number of samples as fast as it can, and sends the average to Data RAM.			

### 37.11.9 CH[0].LIMIT

Address offset: 0x51C

## High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0xFFFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>										
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	LOW		[-32768 to +32767]	Low level limit																											
B	RW	HIGH		[-32768 to +32767]	High level limit																											

## 37.11.10 CH[1].PSEL.P

Address offset: 0x520

Input positive pin selection for CH[1]

### **37.11.11 CH[1].PSELN**

Address offset: 0x524

### Input negative pin selection for CH[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				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
		AnalogInput1	2	AIN1	
		AnalogInput2	3	AIN2	
		AnalogInput3	4	AIN3	
		AnalogInput4	5	AIN4	
		AnalogInput5	6	AIN5	
		AnalogInput6	7	AIN6	
		AnalogInput7	8	AIN7	
		VDD	9	VDD	

### 37.11.12 CH[1].CONFIG

Address offset: 0x528

Input configuration for CH[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	G F E E E D C C C B B A A				
<b>Reset 0x00020000</b>	<b>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A RW RESP					Positive channel resistor control
		Bypass	0	Bypass resistor ladder	
		Pulldown	1	Pull-down to GND	
		Pullup	2	Pull-up to VDD	
		VDD1_2	3	Set input at VDD/2	
B RW RESN					Negative channel resistor control
		Bypass	0	Bypass resistor ladder	
		Pulldown	1	Pull-down to GND	
		Pullup	2	Pull-up to VDD	
		VDD1_2	3	Set input at VDD/2	
C RW GAIN					Gain control
		Gain1_6	0	1/6	
		Gain1_5	1	1/5	
		Gain1_4	2	1/4	
		Gain1_3	3	1/3	
		Gain1_2	4	1/2	
		Gain1	5	1	
		Gain2	6	2	
		Gain4	7	4	
D RW REFSEL					Reference control
		Internal	0	Internal reference (0.6 V)	
		VDD1_4	1	VDD/4 as reference	
E RW TACQ					Acquisition time, the time the ADC uses to sample the input voltage
		3us	0	3 us	
		5us	1	5 us	
		10us	2	10 us	
		15us	3	15 us	
		20us	4	20 us	
		40us	5	40 us	
F RW MODE					Enable differential mode
		SE	0	Single ended, PSELN will be ignored, negative input to ADC shorted to GND	
		Diff	1	Differential	
G RW BURST					Enable burst mode
		Disabled	0	Burst mode is disabled (normal 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								G		F	E	E	E			D	C	C	C		B	B		A	A								
Reset 0x00020000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value																											Description			
		Enabled	1																										Burst mode is enabled. SAADC takes $2^{^{\text{OVERSAMPLE}}}$ number of samples as fast as it can, and sends the average to Data RAM.				

### 37.11.13 CH[1].LIMIT

Address offset: 0x52C

#### High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0xFFFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>										
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																												
A	RW	LOW		[-32768 to +32767]	Low level limit																												
B	RW	HIGH		[-32768 to +32767]	High level limit																												

## 37.11.14 CH[2].PSEL.P

Address offset: 0x530

Input positive pin selection for CH[2]

## 37.11.15 CH[2].PSELN

Address offset: 0x534

Input negative pin selection for CH[2]

## **37.11.16 CH[2].CONFIG**

Address offset: 0x538

## Input configuration for CH[2]

### **37.11.17 CH[2].LIMIT**

Address offset: 0x53C

#### High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0xFFFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>										
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	LOW		[-32768 to +32767]	Low level limit																											
B	RW	HIGH		[-32768 to +32767]	High level limit																											

37.11.18 CH[3].PSEL.P

Address offset: 0x540

Input positive pin selection for CH[3]

37.11.19 CH[3].PSELN

Address offset: 0x544

Input negative pin selection for CH[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
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value		Description																										
A	RW	PSELN				Analog negative input, enables differential channel																										
		NC		0		Not connected																										
		AnalogInput0		1		AIN0																										
		AnalogInput1		2		AIN1																										
		AnalogInput2		3		AIN2																										
		AnalogInput3		4		AIN3																										
		AnalogInput4		5		AIN4																										
		AnalogInput5		6		AIN5																										
		AnalogInput6		7		AIN6																										
		AnalogInput7		8		AIN7																										
		VDD		9		VDD																										

## **37.11.20 CH[3].CONFIG**

Address offset: 0x548

### Input configuration for CH[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												G	F	E	E	E	D	C	C	C	B	B	A	A										
Reset	0x00020000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value																								Description							
A	RW	RESP																									Positive channel resistor control							

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				G	F	E	E	E	E	D	C	C	C	B	B	A	A																		
Reset 0x00020000				0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																														
		Bypass	0		Bypass resistor ladder																														
		Pulldown	1		Pull-down to GND																														
		Pullup	2		Pull-up to VDD																														
		VDD1_2	3		Set input at VDD/2																														
<b>B</b>	<b>RW</b>	<b>RESN</b>			Negative channel resistor control																														
		Bypass	0		Bypass resistor ladder																														
		Pulldown	1		Pull-down to GND																														
		Pullup	2		Pull-up to VDD																														
		VDD1_2	3		Set input at VDD/2																														
<b>C</b>	<b>RW</b>	<b>GAIN</b>			Gain control																														
		Gain1_6	0		1/6																														
		Gain1_5	1		1/5																														
		Gain1_4	2		1/4																														
		Gain1_3	3		1/3																														
		Gain1_2	4		1/2																														
		Gain1	5		1																														
		Gain2	6		2																														
		Gain4	7		4																														
<b>D</b>	<b>RW</b>	<b>REFSEL</b>			Reference control																														
		Internal	0		Internal reference (0.6 V)																														
		VDD1_4	1		VDD/4 as reference																														
<b>E</b>	<b>RW</b>	<b>TACQ</b>			Acquisition time, the time the ADC uses to sample the input voltage																														
		3us	0		3 us																														
		5us	1		5 us																														
		10us	2		10 us																														
		15us	3		15 us																														
		20us	4		20 us																														
		40us	5		40 us																														
<b>F</b>	<b>RW</b>	<b>MODE</b>			Enable differential mode																														
		SE	0		Single ended, PSELN will be ignored, negative input to ADC shorted to GND																														
		Diff	1		Differential																														
<b>G</b>	<b>RW</b>	<b>BURST</b>			Enable burst mode																														
		Disabled	0		Burst mode is disabled (normal operation)																														
		Enabled	1		Burst mode is enabled. SAADC takes 2^OVERSAMPLE number of samples as fast as it can, and sends the average to Data RAM.																														

### 37.11.21 CH[3].LIMIT

Address offset: 0x54C

High/low limits for event monitoring a channel

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14
------------	--	--	--	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	PSEL			Analog positive input channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD

### 37.11.23 CH[4].PSELN

Address offset: 0x554

Input negative pin selection for CH[4]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	PSEL			Analog negative input, enables differential channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD

### 37.11.24 CH[4].CONFIG

Address offset: 0x558

Input configuration for CH[4]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	G F E E E D C C C B B A A				
Reset 0x00020000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	RESP			Positive channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
B	RW	RESN			Negative channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
C	RW	GAIN			Gain control
			Gain1_6	0	1/6
			Gain1_5	1	1/5

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id															G	F	E	E	E	D	C	C	C	B	B	A	A							
Reset 0x000020000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																												
		Gain1_4	2		1/4																													
		Gain1_3	3		1/3																													
		Gain1_2	4		1/2																													
		Gain1	5		1																													
		Gain2	6		2																													
		Gain4	7		4																													
D	RW	REFSEL				Reference control																												
		Internal	0			Internal reference (0.6 V)																												
		VDD1_4	1			VDD/4 as reference																												
E	RW	TACQ				Acquisition time, the time the ADC uses to sample the input voltage																												
		3us	0			3 us																												
		5us	1			5 us																												
		10us	2			10 us																												
		15us	3			15 us																												
		20us	4			20 us																												
		40us	5			40 us																												
F	RW	MODE				Enable differential mode																												
		SE	0			Single ended, PSELN will be ignored, negative input to ADC shorted to GND																												
		Diff	1			Differential																												
G	RW	BURST				Enable burst mode																												
		Disabled	0			Burst mode is disabled (normal operation)																												
		Enabled	1			Burst mode is enabled. SAADC takes 2^OVERSAMPLE number of samples as fast as it can, and sends the average to Data RAM.																												

### 37.11.25 CH[4].LIMIT

Address offset: 0x55C

High/low limits for event monitoring a channel

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																																		
Reset 0x7FFF8000			0	1	1	1	1	1	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	LOW			[ -32768 to +32767 ]	Low level limit																												
B	RW	HIGH			[ -32768 to +32767 ]	High level limit																												

### 37.11.26 CH[5].PSEL

Address offset: 0x560

Input positive pin selection for CH[5]

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																																		
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																												
A	RW	PSEL				Analog positive input channel																												
		NC	0			Not connected																												
		AnalogInput0	1			AIN0																												
		AnalogInput1	2			AIN1																												
		AnalogInput2	3			AIN2																												
		AnalogInput3	4			AIN3																												
		AnalogInput4	5			AIN4																												
		AnalogInput5	6			AIN5																												

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
		AnalogInput6	7	AIN6	
		AnalogInput7	8	AIN7	
		VDD	9	VDD	

### 37.11.27 CH[5].PSELN

Address offset: 0x564

Input negative pin selection for CH[5]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	PSELN			Analog negative input, enables differential channel
		NC	0	Not connected	
		AnalogInput0	1	AIN0	
		AnalogInput1	2	AIN1	
		AnalogInput2	3	AIN2	
		AnalogInput3	4	AIN3	
		AnalogInput4	5	AIN4	
		AnalogInput5	6	AIN5	
		AnalogInput6	7	AIN6	
		AnalogInput7	8	AIN7	
		VDD	9	VDD	

### 37.11.28 CH[5].CONFIG

Address offset: 0x568

Input configuration for CH[5]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	G F E E E D C C C B B A A				
Reset 0x00020000	0 0 0 0 0 0 0 0 0 0 0 1 0				
Id	RW	Field	Value Id	Value	Description
A	RW	RESP			Positive channel resistor control
		Bypass	0	Bypass resistor ladder	
		Pulldown	1	Pull-down to GND	
		Pullup	2	Pull-up to VDD	
		VDD1_2	3	Set input at VDD/2	
B	RW	RESN			Negative channel resistor control
		Bypass	0	Bypass resistor ladder	
		Pulldown	1	Pull-down to GND	
		Pullup	2	Pull-up to VDD	
		VDD1_2	3	Set input at VDD/2	
C	RW	GAIN			Gain control
		Gain1_6	0	1/6	
		Gain1_5	1	1/5	
		Gain1_4	2	1/4	
		Gain1_3	3	1/3	
		Gain1_2	4	1/2	
		Gain1	5	1	
		Gain2	6	2	
		Gain4	7	4	
D	RW	REFSEL			Reference control
		Internal	0	Internal reference (0.6 V)	

### 37.11.29 CH[5].LIMIT

Address offset: 0x56C

## High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x7FFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>										
Id	RW	Field	Value Id	Value	Description																											
A	RW	LOW		[-32768 to +32767]	Low level limit																											
B	RW	HIGH		[-32768 to +32767]	High level limit																											

37.11.30 CH[6].PSEL.P

Address offset: 0x570

Input positive pin selection for CH[6]

37.11.31 CH[6].PSELN

Address offset: 0x574

Input negative pin selection for CH[6]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A A A A A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	PSELN			Analog negative input, enables differential channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD

### 37.11.32 CH[6].CONFIG

Address offset: 0x578

Input configuration for CH[6]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	G F E E E D C C C B B A A				
<b>Reset 0x00020000</b>	<b>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	RESP			Positive channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
B	RW	RESN			Negative channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
C	RW	GAIN			Gain control
			Gain1_6	0	1/6
			Gain1_5	1	1/5
			Gain1_4	2	1/4
			Gain1_3	3	1/3
			Gain1_2	4	1/2
			Gain1	5	1
			Gain2	6	2
			Gain4	7	4
D	RW	REFSEL			Reference control
			Internal	0	Internal reference (0.6 V)
			VDD1_4	1	VDD/4 as reference
E	RW	TACQ			Acquisition time, the time the ADC uses to sample the input voltage
			3us	0	3 us
			5us	1	5 us
			10us	2	10 us
			15us	3	15 us
			20us	4	20 us
			40us	5	40 us
F	RW	MODE			Enable differential mode
			SE	0	Single ended, PSELN will be ignored, negative input to ADC shorted to GND
			Diff	1	Differential

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id												G	F	E	E	E	D	C	C	C	B	B	A	A									
Reset 0x00020000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value																											Description			
G	RW	BURST																											Enable burst mode				
																														Burst mode is disabled (normal operation)			
																													Burst mode is enabled. SAADC takes $2^{OVERSAMPLE}$ number of samples as fast as it can, and sends the average to Data RAM.				
																														Enabled			
																													Disabled				

### **37.11.33 CH[6].LIMIT**

Address offset: 0x57C

#### High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
<b>Reset 0xFFFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>									
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	LOW		[-32768 to +32767]	Low level limit																											
B	RW	HIGH		[-32768 to +32767]	High level limit																											

37.11.34 CH[7].PSEL.P

Address offset: 0x580

Input positive pin selection for CH[7]

37.11.35 CH[7].PSELN

Address offset: 0x584

Input negative pin selection for CH[7]

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																											A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																										
A	RW	PSELN					Analog negative input, enables differential channel																									
			NC		0		Not connected																									
			AnalogInput0		1		AIN0																									
			AnalogInput1		2		AIN1																									
			AnalogInput2		3		AIN2																									
			AnalogInput3		4		AIN3																									
			AnalogInput4		5		AIN4																									
			AnalogInput5		6		AIN5																									

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Id	A A A A A					
Reset 0x00000000	0 0					
Id	RW	Field	Value	Id	Value	Description
		AnalogInput6	7			AIN6
		AnalogInput7	8			AIN7
		VDD	9			VDD

### 37.11.36 CH[7].CONFIG

Address offset: 0x588

Input configuration for CH[7]

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0					
Id	G F E E E D C C C B B A A					
Reset 0x000020000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0					
Id	RW	Field	Value	Id	Value	Description
A	RW	RESP				Positive channel resistor control
		Bypass	0			Bypass resistor ladder
		Pulldown	1			Pull-down to GND
		Pullup	2			Pull-up to VDD
		VDD1_2	3			Set input at VDD/2
B	RW	RESN				Negative channel resistor control
		Bypass	0			Bypass resistor ladder
		Pulldown	1			Pull-down to GND
		Pullup	2			Pull-up to VDD
		VDD1_2	3			Set input at VDD/2
C	RW	GAIN				Gain control
		Gain1_6	0			1/6
		Gain1_5	1			1/5
		Gain1_4	2			1/4
		Gain1_3	3			1/3
		Gain1_2	4			1/2
		Gain1	5			1
		Gain2	6			2
		Gain4	7			4
D	RW	REFSEL				Reference control
		Internal	0			Internal reference (0.6 V)
		VDD1_4	1			VDD/4 as reference
E	RW	TACQ				Acquisition time, the time the ADC uses to sample the input voltage
		3us	0			3 us
		5us	1			5 us
		10us	2			10 us
		15us	3			15 us
		20us	4			20 us
		40us	5			40 us
F	RW	MODE				Enable differential mode
		SE	0			Single ended, PSELN will be ignored, negative input to ADC shorted to GND
		Diff	1			Differential
G	RW	BURST				Enable burst mode
		Disabled	0			Burst mode is disabled (normal operation)
		Enabled	1			Burst mode is enabled. SAADC takes 2^OVERSAMPLE number of samples as fast as it can, and sends the average to Data RAM.

### 37.11.37 CH[7].LIMIT

Address offset: 0x58C

High/low limits for event monitoring a channel

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x7FFF8000</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>									
Id	RW	Field	Value Id	Value	Description																											
A	RW	LOW		[-32768 to +32767]	Low level limit																											
B	RW	HIGH		[-32768 to +32767]	High level limit																											

## **37.11.38 RESOLUTION**

Address offset: 0x5F0

## Resolution configuration

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	VAL			Set the resolution																											
		8bit	0	8 bit																												
		10bit	1	10 bit																												
		12bit	2	12 bit																												
		14bit	3	14 bit																												

## **37.11.39 OVERSAMPLE**

Address offset: 0x5F4

Oversampling configuration. OVERSAMPLE should not be combined with SCAN. The RESOLUTION is applied before averaging, thus for high OVERSAMPLE a higher RESOLUTION should be used.

## **37.11.40 SAMPLERATE**

Address offset: 0x5F8

Controls normal or continuous sample rate

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CC		[80..2047]	Capture and compare value. Sample rate is 16 MHz/CC																											
B	RW	MODE			Select mode for sample rate control																											
		Task	0		Rate is controlled from SAMPLE task																											
		Timers	1		Rate is controlled from local timer (use CC to control the rate)																											

### 37.11.41 RESULT.PTR

Address offset: 0x62C

Data pointer

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

### 37.11.42 RESULT.MAXCNT

Address offset: 0x630

Maximum number of buffer words to transfer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	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				
Id	RW	Field	Value	Id	Value	Description																										
A	RW	MAXCNT				Maximum number of buffer words to transfer																										

### 37.11.43 RESULT.AMOUNT

Address offset: 0x634

Number of buffer words transferred since last START

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	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				
Id	RW	Field	Value	Id	Value	Description																										
A	R	AMOUNT				Number of buffer words transferred since last START. This register can be read after an END or STOPPED event.																										

## 37.12 Electrical specification

### 37.12.1 SAADC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
DNL	Differential non-linearity, 10-bit resolution	-0.95	<1		LSB
INL	Integral non-linearity, 10-bit resolution	1			LSB
V <sub>OS</sub>	Differential offset error (calibrated), 10-bit resolution <sup>a</sup>	+2			LSB
C <sub>EG</sub>	Gain error temperature coefficient	0.02			%/°C
f <sub>SAMPLE</sub>	Maximum sampling rate	200			kHz
t <sub>ACQ,10k</sub>	Acquisition time (configurable), source Resistance <= 10kOhm	3			μs
t <sub>ACQ,40k</sub>	Acquisition time (configurable), source Resistance <= 40kOhm	5			μs
t <sub>ACQ,100k</sub>	Acquisition time (configurable), source Resistance <= 100kOhm	10			μs
t <sub>ACQ,200k</sub>	Acquisition time (configurable), source Resistance <= 200kOhm	15			μs
t <sub>ACQ,400k</sub>	Acquisition time (configurable), source Resistance <= 400kOhm	20			μs
t <sub>ACQ,800k</sub>	Acquisition time (configurable), source Resistance <= 800kOhm	40			μs
t <sub>CONV</sub>	Conversion time	<2			μs
I <sub>ADC,CONV</sub>	ADC current during ACQuisition and CONVersion	700			μA
I <sub>ADC,IDLE</sub>	Idle current, when not sampling, excluding clock sources and regulator base currents <sup>33</sup>	<5			μA

<sup>a</sup> Digital output code at zero volt differential input.

<sup>33</sup> When t<sub>ACQ</sub> is 10us or longer, and if DC/DC is active, it will be allowed to work in refresh mode if no other resource is requiring a high quality power supply from 1V3. If t<sub>ACQ</sub> is smaller than 10us and DC/DC is active,

Symbol	Description	Min.	Typ.	Max.	Units
$E_{G1/6}$	Error <sup>b</sup> for Gain = 1/6	-3	3	3	%
$E_{G1/4}$	Error <sup>b</sup> for Gain = 1/4	-3	3	3	%
$E_{G1/2}$	Error <sup>b</sup> for Gain = 1/2	-3	4	4	%
$E_{G1}$	Error <sup>b</sup> for Gain = 1	-3	4	4	%
$C_{SAMPLE}$	Sample and hold capacitance at maximum gain <sup>34</sup>	2.5		pF	
$R_{INPUT}$	Input resistance	>1		MΩ	
$E_{NOB}$	Effective number of bits, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 kspS	9		Bit	
$S_{NDR}$	Peak signal to noise and distortion ratio, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 kspS	56		dB	
$S_{FDR}$	Spurious free dynamic range, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 kspS	70		dBc	
$R_{LADDER}$	Ladder resistance	160		kΩ	

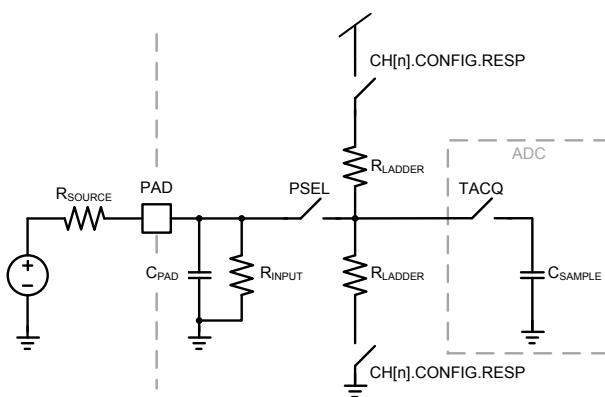


Figure 105: Model of SAADC input (one channel)

Note: SAADC average current calculation for a given application is based on the sample period, conversion and acquisition time ( $t_{CONV}$  and  $t_{ACQ}$ ) and conversion and idle current ( $I_{ADC,CONV}$  and  $I_{ADC,IDLE}$ ). For example, sampling at 4kHz gives a sample period of 250µs. The average current consumption would then be:

$$I_{AVERAGE} = \left( \frac{(t_{CONV} + t_{ACQ})}{250} \right) (I_{ADC,CONV}) + \left( \frac{250 - (t_{CONV} + t_{ACQ})}{250} \right) (I_{ADC,IDLE})$$

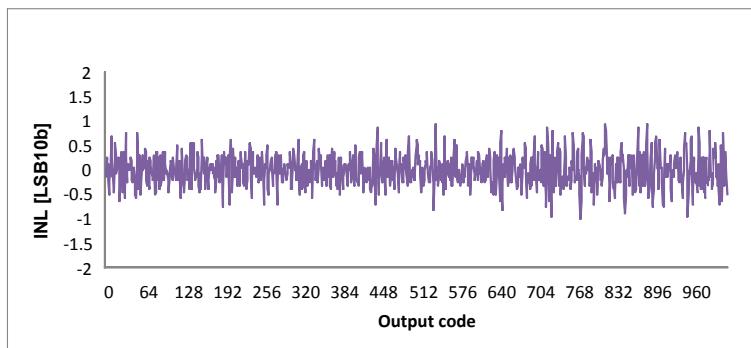


Figure 106: ADC INL vs Output Code

refresh mode will not be allowed, and it will remain in normal mode from the START task to the STOPPED event.

So depending on  $t_{ACQ}$  and other resources' needs, the appropriate base current needs to be taken into account.

<sup>b</sup> Does not include temperature drift

<sup>34</sup> Maximum gain corresponds to highest capacitance.

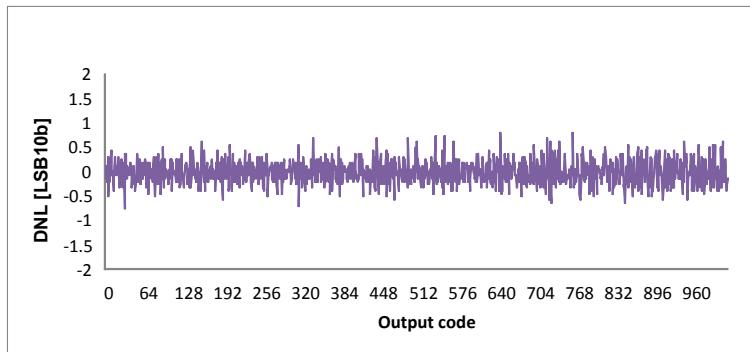


Figure 107: ADC DNL vs Output Code

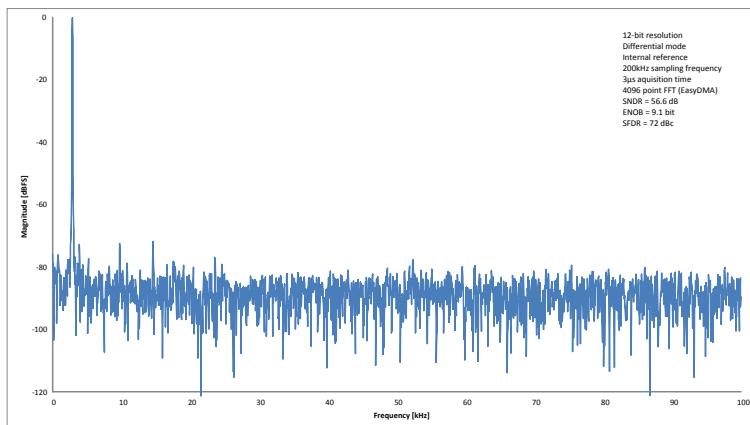


Figure 108: FFT of a 2.8 kHz sine at 200 ksps ()

### 37.13 Performance factors

Clock jitter, affecting sample timing accuracy, and circuit noise can affect ADC performance.

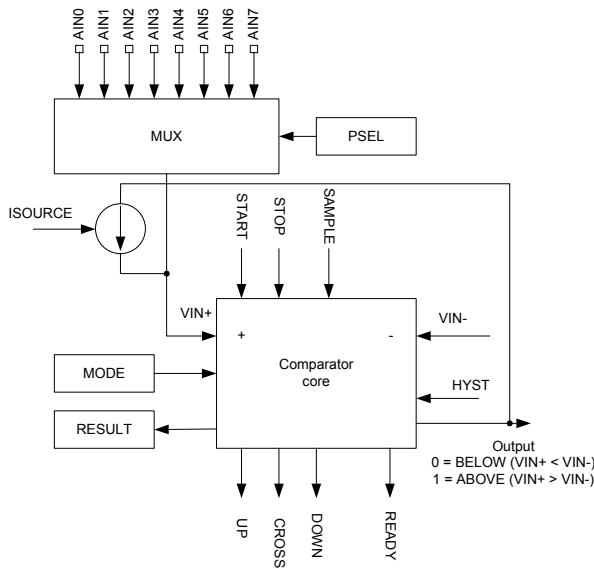
Jitter can be between START tasks or from START task to acquisition. START timer accuracy and startup times of regulators and references will contribute to variability. Sources of circuit noise may include CPU activity and the DC/DC regulator. Best ADC performance is achieved using START timing based on the TIMER module, HFXO clock source, and Constant Latency mode.

## 38 COMP — Comparator

The Comparator (COMP) compares the input voltage ( $V_{IN+}$ ) that is derived from an analog input pin selected via the PSEL register against a second input voltage ( $V_{IN-}$ ) that can be derived from multiple sources depending on operation mode.

Listed here are the main features for COMP:

- Input range from 0 V to VDD
- Single-ended mode
  - Fully flexible hysteresis using a 64-level reference ladder
- Differential mode
  - Configurable 50 mV hysteresis
- Reference inputs:
  - VDD
  - External reference from AIN0 to AIN1 (between 0 V and VDD)
  - Internal references 1.2 V, 1.8 V and 2.4 V
- Three operation modes: low power, normal and high-speed
- Single-pin capacitive sensor support
- Event generation on output changes
  - UP event on  $V_{IN-} > V_{IN+}$
  - DOWN event on  $V_{IN-} < V_{IN+}$
  - CROSS event on  $V_{IN+}$  and  $V_{IN-}$  crossing
  - READY event on core and internal reference (if used) ready



**Figure 109: Comparator overview**

**Important:** COMP cannot be used (STARTed) at the same time as LPCOMP. Only one comparator can be used at a time.

The COMP is started by triggering the START task, and stopped by triggering the STOP task. After a start-up time of  $t_{COMP,START}$ <sup>35</sup> the COMP will generate a READY event to indicate that the comparator is ready

<sup>35</sup> See  $t_{PROPDLY,LP}$ ,  $t_{PROPDLY,N}$ ,  $t_{PROPDLY,HS}$ ,  $I_{COMP,LP}$ ,  $I_{COMP,N}$  and  $I_{COMP,HS}$  in [Electrical parameters](#) for more information about COMP speed and power characteristics related to these different modes.

to use and the output of the COMP is correct. When the COMP module is started, events will be generated every time VIN+ crosses VIN-.

VIN- can be derived directly from AIN0 or AIN1 in differential mode, or VREF in single-ended mode. VUP and VDOWN thresholds can be set to implement a hysteresis on VIN- using the Reference Ladder. VREF can be derived from VDD, AIN0, AIN1 or internal 1.2V, 1.8V and 2.4V references.

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

An optional hysteresis on VIN+ and VIN- can be enabled when the module is used in differential mode through the HYST register. In single ended mode the two reference ladders (VUP and VDOWN, see [Figure 112: Comparator in single-ended mode](#) on page 394) will be used instead of the hysteresis mechanism configured in HYST.

This hysteresis is in the order of magnitude of 50 mV, and shall prevent noise on the signal to create unwanted events. See [Figure 113: Hysteresis example where VIN+ starts below VUP](#) on page 394 for illustration of the effect of an active hysteresis on a noisy input signal.

The COMP can be configured to operate in two main operation modes, differential mode and single-ended mode, see MODE register for more information.

The COMP can, for both main operation modes, operate in different speed and power consumption modes, see MODE register. High-speed mode will consume more power compared to low-power mode, and low-power mode will result in slower response time compared to high-speed mode.

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

A selectable current can be applied (ISOURCE register) on the currently selected AINx line. Enabling the block creates a feedback path around the comparator, forming a relaxation oscillator. The circuit will sink current from VIN+ when the comparator output is high, and source current into VIN+ when the comparator output is low. The frequency of the oscillator is dependent on the capacitance at the analog input pin, the reference voltages and the value of the current source. In this mode, only a capacitive sensor needs to be attached between the analog input pin and ground. With a selected current of 10  $\mu$ A, VUP-VDOWN equal to 1 V, and an external capacity of typically 10 pF, the resulting oscillation frequency is around 500 kHz.

The frequency of the oscillator can be calculated as

$$f_{\text{OSC}} = I_{\text{SOURCE}} / (2C \cdot (V_{\text{UP}} - V_{\text{DOWN}}))$$

## 38.1 Shared resources

The COMP shares analog resources with the SAADC and LPCOMP peripherals.

While it is possible to use SAADC at the same time as COMP or LPCOMP, COMP and LPCOMP are mutually exclusive: enabling one will automatically disable the other. In addition, when using SAADC and COMP or LPCOMP simultaneously, it is not possible to select the same analog input pin for both modules.

**Important:** The COMP peripheral shall not be disabled (by writing to the ENABLE register) before the peripheral has stopped. Failing to do so may result in unpredictable behaviour.

## 38.2 Differential mode

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

In this mode, the impedance on VIN-'s signal path is equal to the impedance on VIN+'s signal path. See  $Z_{\text{COMPVINP}}$  and  $Z_{\text{COMPVINND}}$  for more information. In differential mode, the PSEL, MODE and EXTREFSEL registers must be configured before the COMP is enabled via the ENABLE register. When HYST is turned on while in this mode, the Output of the comparator (and associated events) will change from ABOVE to BELOW whenever VIN+ becomes smaller than ( $V_{\text{IN+}} - (V_{\text{DIFFHYST}} / 2)$ ). Similarly, it will change from BELOW to ABOVE whenever VIN+ becomes larger than ( $V_{\text{IN+}} + (V_{\text{DIFFHYST}} / 2)$ ), as illustrated in [Figure 111: Hysteresis enabled in differential mode](#) on page 393.

**Restriction:** Depending on the device, not all the analog inputs may be available for each MUX.

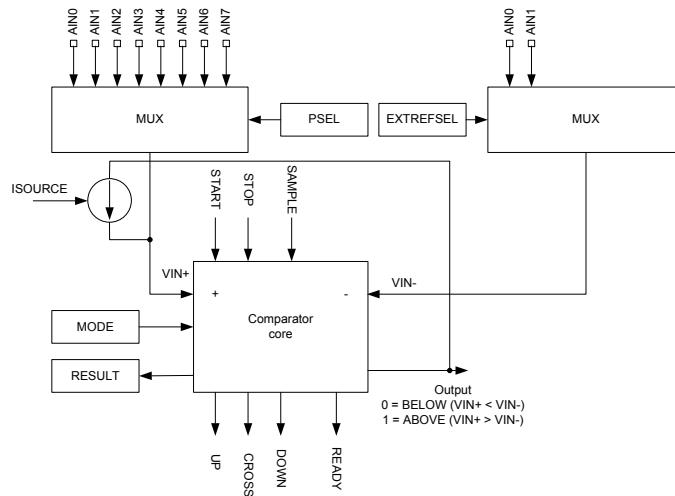


Figure 110: Comparator in differential mode

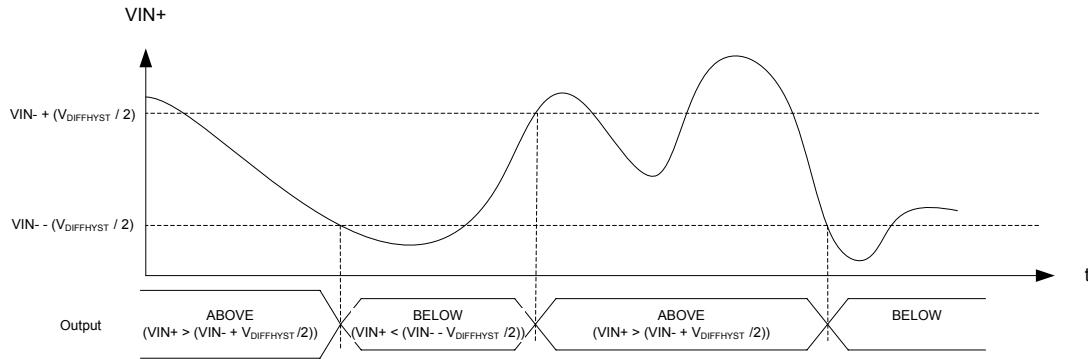


Figure 111: Hysteresis enabled in differential mode

### 38.3 Single-ended mode

In single-ended mode, VIN- is derived from the Reference Ladder.

The Reference Ladder uses the reference voltage VREF to derive two new voltage references, VUP and VDOWN. VUP and VDOWN are configured via THUP and THDOWN in the TH register. VREF can be sourced from any of the available references sources as illustrated in [Figure 112: Comparator in single-ended mode](#) on page 394. This is configured via EXTREFSEL and REFSEL.

When the comparator core detects that  $VIN+ > VIN-$ , i.e. ABOVE as per the RESULT register, VIN- will switch to VDOWN. When VIN- falls below VIN- again, VIN- will be switched back to VUP. By specifying VUP larger than VDOWN, a hysteresis as illustrated in [Figure 113: Hysteresis example where VIN+ starts below VUP](#) on page 394 and [Figure 114: Hysteresis example where VIN+ starts above VUP](#) on page 395 can be generated. In single-ended mode, the PSEL, MODE, EXTREFSEL, REFSEL and TH registers must be configured before the COMP is enabled via the ENABLE register.

**Restriction:** Depending on the device, not all the analog inputs may be available for each MUX.

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

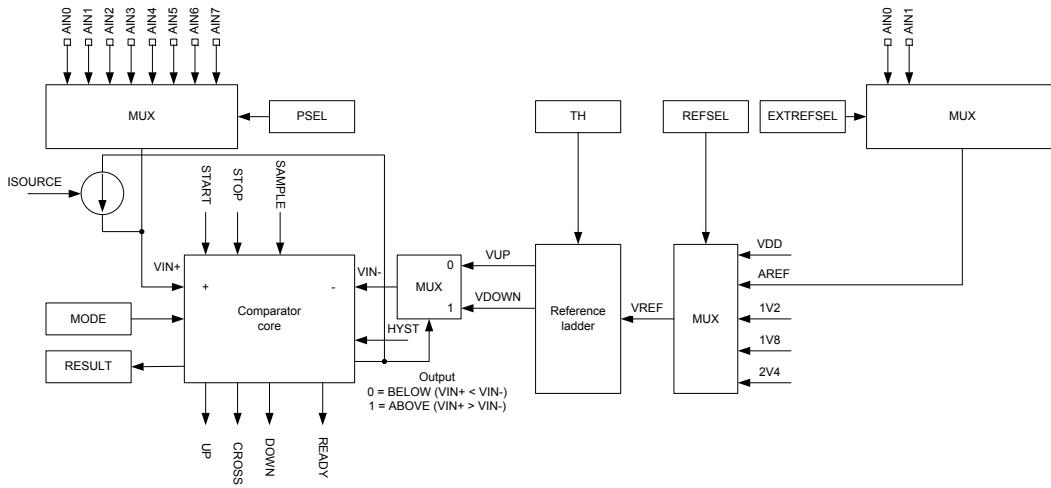


Figure 112: Comparator in single-ended mode

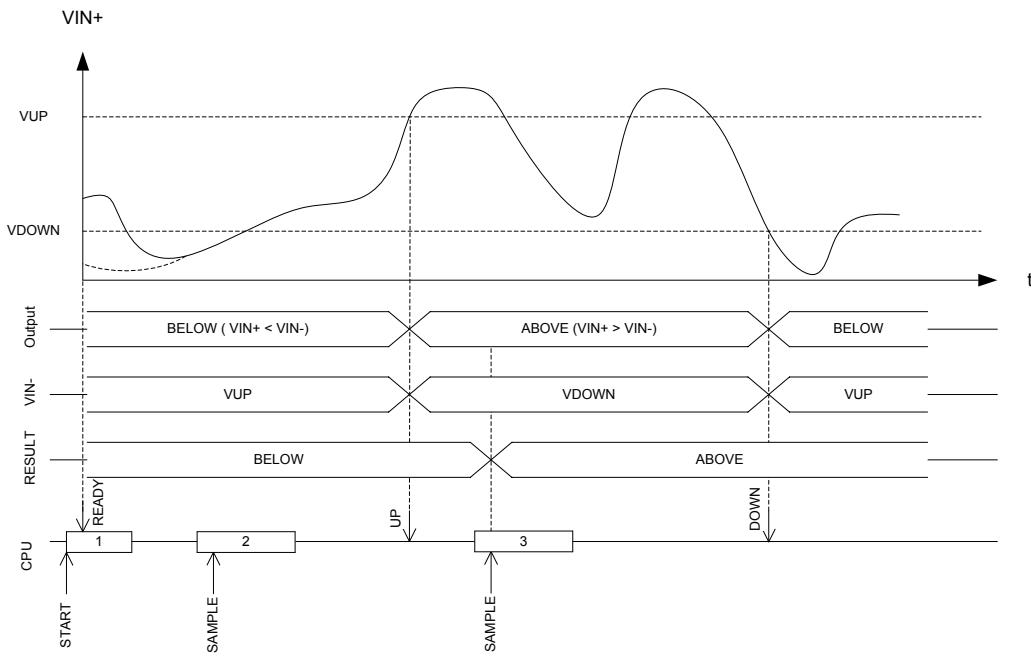


Figure 113: Hysteresis example where  $V_{IN+}$  starts below  $V_{UP}$

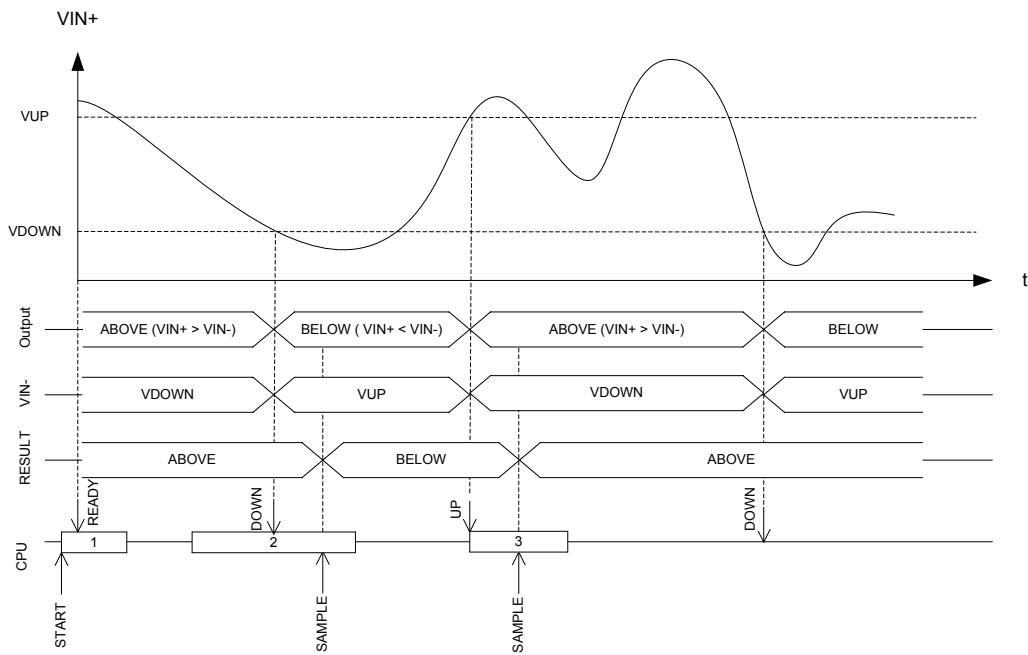


Figure 114: Hysteresis example where  $V_{IN+}$  starts above  $V_{UP}$

## 38.4 Pin configuration

The user can use the PSEL register to select one of the analog input pins,  $A_{IN0}$  through  $A_{IN7}$ , as input  $V_{IN+}$ .

See [Figure 112: Comparator in single-ended mode](#) on page 394. Similarly the user can use the EXTREFSEL register to select one of the  $A_{INx}$  analog input pins as reference input, in case AREF is selected in REFSEL. The selected analog pins will be acquired by the COMP when it is enabled via the ENABLE register.

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

## 38.5 Registers

Table 91: Instances

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

Table 92: Register Overview

Register	Offset	Description
TASKS_START	0x000	Start comparator
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	COMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
<b>SHORTS</b>	0x200	Shortcut register
<b>INTEN</b>	0x300	Enable or disable interrupt
<b>INTENSET</b>	0x304	Enable interrupt

Register	Offset	Description
<i>INTENCLR</i>	0x308	Disable interrupt
<i>RESULT</i>	0x400	Compare result
<i>ENABLE</i>	0x500	COMP enable
<i>PSEL</i>	0x504	Pin select
<i>REFSEL</i>	0x508	Reference source select
<i>EXTREFSEL</i>	0x50C	External reference select
<i>TH</i>	0x530	Threshold configuration for hysteresis unit
<i>MODE</i>	0x534	Mode configuration
<i>HYST</i>	0x538	Comparator hysteresis enable
<i>ISOURCE</i>	0x53C	Current source select on analog input

### 38.5.1 SHORTS

Address offset: 0x200

Shortcut register

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

### 38.5.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	C	B	A																												
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	READY				Enable or disable interrupt for READY event																										
						See <a href="#">EVENTS_READY</a>																										
		Disabled	0			Disable																										
		Enabled	1			Enable																										
B	RW	DOWN				Enable or disable interrupt for DOWN event																										
						See <a href="#">EVENTS_DOWN</a>																										

Bit number	31	30	29	28	27	26	25	24	23	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																								Description					
C			Disabled																								Disable					
C			Enabled																								Enable					
C			UP																								Enable or disable interrupt for UP event					
D																											See <a href="#">EVENTS_UP</a>					
D			Disabled																								Disable					
D			Enabled																								Enable					
D			CROSS																								Enable or disable interrupt for CROSS event					
D																											See <a href="#">EVENTS_CROSS</a>					
D			Disabled																								Disable					
D			Enabled																								Enable					

### **38.5.3 INTENSET**

Address offset: 0x304

## Enable interrupt

### **38.5.4 INTENCLR**

Address offset: 0x308

Disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																											D	C	B	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	READY				Write '1' to Disable interrupt for READY event																										
						See <a href="#">EVENTS_READY</a>																										
		Clear	1			Disable																										

### **38.5.5 RESULT**

Address offset: 0x400

## Compare result

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

### **38.5.6 ENABLE**

Address offset: 0x500

### COMP enable

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

### **38.5.7 PSEL**

Address offset: 0x504

## Pin select

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PSEL			Analog pin select																											

## 38.5.8 REFSEL

Address offset: 0x508

## Reference source select

### **38.5.9 EXTREFSEL**

Address offset: 0x50C

## External reference select

### **38.5.10 TH**

Address offset: 0x530

## Threshold configuration for hysteresis unit

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

### **38.5.11 MODE**

Address offset: 0x534

## Mode configuration

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

### 38.5.12 HYST

Address offset: 0x538

Comparator hysteresis enable

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

### 38.5.13 ISOURCE

Address offset: 0x53C

Current source select on analog input

Bit number	31 30 29 28 27 26 25 24 23 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>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	ISOURCE			Comparator hysteresis
		Off	0		Current source disabled
		Ien2mA5	1		Current source enabled (+/- 2.5 uA)
		Ien5mA	2		Current source enabled (+/- 5 uA)
		Ien10mA	3		Current source enabled (+/- 10 uA)

## 38.6 Electrical specification

### 38.6.1 COMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$I_{COMP,L}$	Core run current in low power mode	2			µA
$I_{COMP,N}$	Core run current in normal mode	5			µA
$I_{COMP,HS}$	Core run current in high-speed mode	10			µA
$t_{PROPDLY,L}$	Propagation delay, low power mode <sup>a</sup>	0.6			µS
$t_{PROPDLY,N}$	Propagation delay, normal mode <sup>a</sup>	0.2			µS
$t_{PROPDLY,HS}$	Propagation delay, high-speed mode <sup>a</sup>	0.1			µS
$I_{SOURCE}$	Configurable input current provided by the output driven current source.				µA
$I_{SOURCE,A}$		2.5			µA
$I_{SOURCE,B}$		5.0			µA
$I_{SOURCE,C}$		10.0			µA

<sup>a</sup> Propagation delay is with 10mV overdrive.

Symbol	Description	Min.	Typ.	Max.	Units
$V_{DIFFHYST}$	Optional hysteresis applied to differential input		30		mV
$V_{VDD-VREF}$	Required difference between VDD and a selected VREF, $VDD > VREF$		0.3		V
$I_{INT\_REF}$	Current used by the internal bandgap reference when selected as source for VREF		13		$\mu A$
$t_{INT\_REF,START}$	Startup time for the internal bandgap reference		50	80	$\mu s$
$E_{INT\_REF}$	Internal bandgap reference error	-3	3		%
$R_{LADDER}$	Reference ladder resistance, $I_{LADDER} = VREF / R_{LADDER}$		550		$k\Omega$
$V_{INPUTOFFSET}$	Input offset	-10	10		mV
$D_{NL LADDER}$	Differential non-linearity of reference ladder		<0.1		LSB
$t_{COMP,START}$	Startup time for the comparator core		3		$\mu s$

Total comparator run current must be calculated from the  $I_{COMP}$ ,  $I_{INT\_REF}$ ,  $I_{SOURCE}$  and  $I_{LADDER}$  values for a given reference voltage.

## 39 LPCOMP — Low power comparator

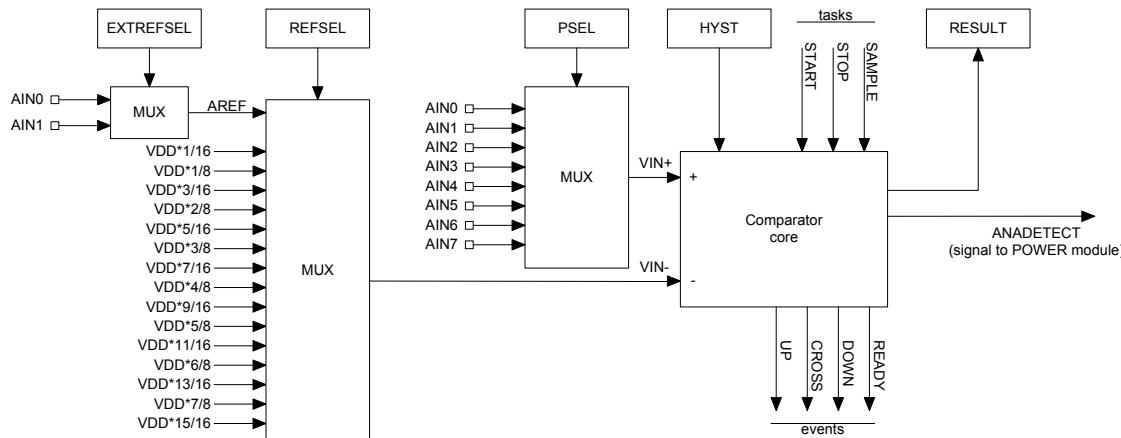
LPCOMP compares an input voltage against a reference voltage.

Listed here are the main features of LPCOMP:

- 0 - VDD input range
- Ultra low power
- Eight input options (AIN0 to AIN7)
- Reference voltage options:
  - Two external analog reference inputs, or
  - 15-level internal reference ladder (VDD/16)
- Optional hysteresis enable on input
- Wakeup source from OFF mode

In System ON, the LPCOMP can generate separate events on rising and falling edges of a signal, or sample the current state of the pin as being above or below the selected reference. The block can be configured to use any of the analog inputs on the device. Additionally, the low power comparator can be used as an analog wakeup source from System OFF or System ON. The comparator threshold can be programmed to a range of fractions of the supply voltage.

**Restriction:** LPCOMP cannot be used (STARTed) at the same time as COMP. Only one comparator can be used at a time.



**Figure 115: Low power comparator**

The wakeup comparator (LPCOMP) compares an input voltage ( $V_{IN+}$ ), which comes from an analog input pin selected via the PSEL register against a reference voltage ( $V_{IN-}$ ) selected via the [REFSEL](#) on page 407 and [EXTREFSEL](#) registers.

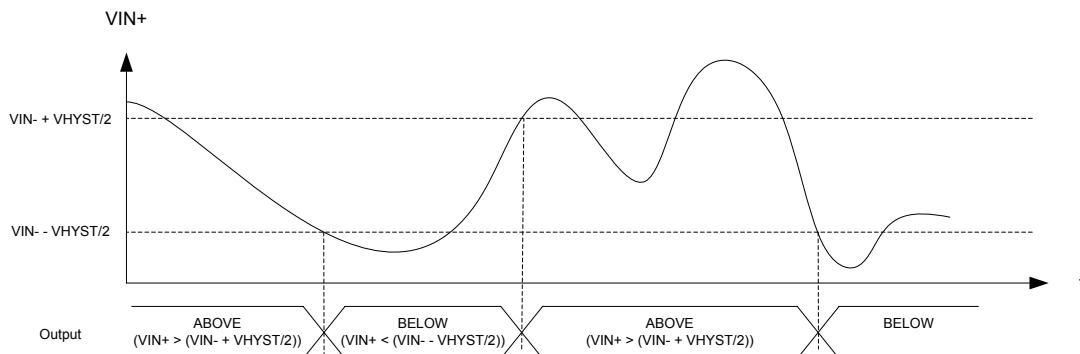
The [PSEL](#), [REFSEL](#), and [EXTREFSEL](#) registers must be configured before the LPCOMP is enabled through the [ENABLE](#) register.

The [HYST](#) register allows enabling an optional hysteresis in the comparator core. This hysteresis is in the order of magnitude of 50 mV, and shall prevent noise on the signal to create unwanted events. See [Figure 116: Effect of hysteresis on a noisy input signal](#) on page 403 for illustration of the effect of an active hysteresis on a noisy input signal. It is disabled by default, and shall be configured before enabling LPCOMP as well.

The LPCOMP is started by triggering the START task. After a start-up time of  $t_{LPCOMP,STARTUP}$  the LPCOMP will generate a READY event to indicate that the comparator is ready to use and the output of the LPCOMP is correct. The LPCOMP will generate events every time  $V_{IN+}$  crosses  $V_{IN-}$ . More specifically, every time  $V_{IN+}$  rises above  $V_{IN-}$  (upward crossing) an UP event is generated along with a CROSS event. Every time  $V_{IN+}$  falls below  $V_{IN-}$  (downward crossing), a DOWN event is generated along with a CROSS event. When

hysteresis is enabled, the upward crossing level becomes ( $VIN_- + VHYST/2$ ), and the downward crossing level becomes ( $VIN_- - VHYST/2$ ).

The LPCOMP is stopped by triggering the STOP task.



**Figure 116: Effect of hysteresis on a noisy input signal**

LPCOMP will be operational in both System ON and System OFF mode when it is enabled through the [ENABLE](#) register. See [POWER — Power supply](#) on page 78 for more information about power modes. Note that it is not allowed to go to System OFF when a READY event is pending to be generated.

All LPCOMP registers, including [ENABLE](#), are classified as retained registers when the LPCOMP is enabled. However, when the device wakes up from System OFF, all LPCOMP registers will be reset.

The LPCOMP can wake up the system from System OFF by asserting the ANADETECT signal. The ANADETECT signal can be derived from any of the event sources that generate the UP, DOWN and CROSS events. In case of wakeup from System OFF, no events will be generated, only the ANADETECT signal. See the ANADETECT register ([ANADETECT](#) on page 407) for more information on how to configure the ANADETECT signal.

The immediate value of the LPCOMP can be sampled to [RESULT](#) on page 406 by triggering the SAMPLE task.

See [RESETREAS](#) on page 85 for more information on how to detect a wakeup from LPCOMP.

## 39.1 Shared resources

The LPCOMP shares resources with other peripherals.

The LPCOMP shares analog resources with SAADC and COMP. While it is possible to use SAADC at the same time as COMP or LPCOMP, COMP and LPCOMP are mutually exclusive: enabling one will automatically disable the other. In addition, when using SAADC and COMP or LPCOMP simultaneously, it is not possible to select the same analog input pin for both modules.

The LPCOMP peripheral shall not be disabled (by writing to the ENABLE register) before the peripheral has been stopped. Failing to do so may result in unpredictable behaviour.

## 39.2 Pin configuration

You can use the LPCOMP.PSEL register to select one of the analog input pins,  $AIN_0$  through  $AIN_7$ , as the analog input pin for the LPCOMP.

See [GPIO — General purpose input/output](#) on page 111 for more information about the pins. Similarly, you can use [EXTREFSEL](#) on page 407 to select one of the analog reference input pins,  $AIN_0$  and  $AIN_1$ , as input for AREF in case AREF is selected in [EXTREFSEL](#) on page 407. The selected analog pins will be acquired by the LPCOMP when it is enabled through [ENABLE](#) on page 406.

### 39.3 Registers

**Table 93: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40013000	LPCOMP	LPCOMP	Low power comparator	

**Table 94: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start comparator
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	LPCOMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
<i>SHORTS</i>	0x200	Shortcut register
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>RESULT</i>	0x400	Compare result
<i>ENABLE</i>	0x500	Enable LPCOMP
<i>PSEL</i>	0x504	Input pin select
<i>REFSEL</i>	0x508	Reference select
<i>EXTREFSEL</i>	0x50C	External reference select
<i>ANADETECT</i>	0x520	Analog detect configuration
<i>HYST</i>	0x538	Comparator hysteresis enable

### **39.3.1 SHORTS**

Address offset: 0x200

## Shortcut register

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

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	D C B A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	READY	Enabled	1	Enable shortcut

### 39.3.2 INTENSET

Address offset: 0x304

Enable interrupt

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

### 39.3.3 INTENCLR

Address offset: 0x308

Disable interrupt

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

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	D C B A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
D RW CROSS					Write '1' to Disable interrupt for CROSS event
					See <a href="#">EVENTS_CROSS</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

### 39.3.4 RESULT

Address offset: 0x400

Compare result

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

### 39.3.5 ENABLE

Address offset: 0x500

Enable LPCOMP

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

### 39.3.6 PSEL

Address offset: 0x504

Input pin select

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

### **39.3.7 REFSEL**

Address offset: 0x508

## Reference select

### **39.3.8 EXTREFSEL**

Address offset: 0x50C

## External reference select

### **39.3.9 ANADETECT**

Address offset: 0x520

## Analog detect configuration

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																									A	A							
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id																								Description						
A	RW	ANADETECT																									Analog detect configuration						
																									Generate ANADETECT on crossing, both upward crossing and downward crossing								
																									Generate ANADETECT on upward crossing only								
																									Generate ANADETECT on downward crossing only								

### **39.3.10 HYST**

Address offset: 0x538

Comparator hysteresis enable

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	HYST	NoHyst	0	Comparator hysteresis enable
			Hyst50mV	1	Comparator hysteresis disabled (typ. 50 mV)

## 39.4 Electrical specification

### 39.4.1 LPCOMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
I <sub>LPC</sub>	Run current for low power comparator	0.5			µA
t <sub>LPCANADET</sub>	Time from VIN crossing (>=50mV above threshold) to ANADETECT signal generated.	5			µs
E <sub>REFLADDER</sub>	Error in reference ladder threshold voltage	-30		30	mV
V <sub>HYST</sub>	Optional hysteresis	30			mV

# 40 WDT — Watchdog timer

A countdown watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up.

The watchdog timer is started by triggering the START task.

The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU. The watchdog is implemented as a down-counter that generates a TIMEOUT event when it wraps over after counting down to 0. When the watchdog timer is started through the START task, the watchdog counter is loaded with the value specified in the CRV register. This counter is also reloaded with the value specified in the CRV register when a reload request is granted.

The watchdog's timeout period is given by:

```
timeout [s] = ( CRV + 1 ) / 32768
```

When started, the watchdog will automatically force the 32.768 kHz RC oscillator on as long as no other 32.768 kHz clock source is running and generating the 32.768 kHz system clock, see chapter [CLOCK — Clock control](#) on page 101.

## 40.1 Reload criteria

The watchdog has eight separate reload request registers, which shall be used to request the watchdog to reload its counter with the value specified in the CRV register. To reload the watchdog counter, the special value 0x6E524635 needs to be written to all enabled reload registers.

One or more RR registers can be individually enabled through the RREN register.

## 40.2 Temporarily pausing the watchdog

By default, the watchdog will be active counting down the down-counter while the CPU is sleeping and when it is halted by the debugger. It is however possible to configure the watchdog to automatically pause while the CPU is sleeping as well as when it is halted by the debugger.

## 40.3 Watchdog reset

A TIMEOUT event will automatically lead to a watchdog reset.

See [Reset](#) on page 82 for more information about reset sources. If the watchdog is configured to generate an interrupt on the TIMEOUT event, the watchdog reset will be postponed with two 32.768 kHz clock cycles after the TIMEOUT event has been generated. Once the TIMEOUT event has been generated, the impending watchdog reset will always be effectuated.

The watchdog must be configured before it is started. After it is started, the watchdog's configuration registers, which comprise registers CRV, RREN, and CONFIG, will be blocked for further configuration.

The watchdog can be reset from several reset sources, see [Reset behavior](#) on page 83.

When the device starts running again, after a reset, or waking up from OFF mode, the watchdog configuration registers will be available for configuration again.

## 40.4 Registers

**Table 95: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40010000	WDT	WDT	Watchdog timer	

**Table 96: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Start the watchdog
EVENTS_TIMEOUT	0x100	Watchdog timeout
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>RUNSTATUS</i>	0x400	Run status
<i>REQSTATUS</i>	0x404	Request status
<i>CRV</i>	0x504	Counter reload value
<i>RREN</i>	0x508	Enable register for reload request registers
<i>CONFIG</i>	0x50C	Configuration register
<i>RR[0]</i>	0x600	Reload request 0
<i>RR[1]</i>	0x604	Reload request 1
<i>RR[2]</i>	0x608	Reload request 2
<i>RR[3]</i>	0x60C	Reload request 3
<i>RR[4]</i>	0x610	Reload request 4
<i>RR[5]</i>	0x614	Reload request 5
<i>RR[6]</i>	0x618	Reload request 6
<i>RR[7]</i>	0x61C	Reload request 7

#### **40.4.1 INTENSET**

Address offset: 0x304

## Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	TIMEOUT				Write '1' to Enable interrupt for TIMEOUT event																										
						See <a href="#">EVENTS_TIMEOUT</a>																										
		Set	1			Enable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										

## 40.4.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	TIMEOUT				Write '1' to Disable interrupt for TIMEOUT event																										
						See <a href="#">EVENTS_TIMEOUT</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										

### 40.4.3 RUNSTATUS

Address offset: 0x400

## Run status

#### **40.4.4 REQSTATUS**

Address offset: 0x404

## Request status

#### **40.4.5 CRV**

Address offset: 0x504

## Counter reload value

## 40.4.6 RREN

Address offset: 0x508

Enable register for reload request registers

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id			H G F E D C B A		
Reset 0x00000001			0 1		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	RR0			Enable or disable RR[0] register
			Disabled	0	Disable RR[0] register
			Enabled	1	Enable RR[0] register
B	RW	RR1			Enable or disable RR[1] register
			Disabled	0	Disable RR[1] register
			Enabled	1	Enable RR[1] register
C	RW	RR2			Enable or disable RR[2] register
			Disabled	0	Disable RR[2] register
			Enabled	1	Enable RR[2] register
D	RW	RR3			Enable or disable RR[3] register
			Disabled	0	Disable RR[3] register
			Enabled	1	Enable RR[3] register
E	RW	RR4			Enable or disable RR[4] register
			Disabled	0	Disable RR[4] register
			Enabled	1	Enable RR[4] register
F	RW	RR5			Enable or disable RR[5] register
			Disabled	0	Disable RR[5] register
			Enabled	1	Enable RR[5] register
G	RW	RR6			Enable or disable RR[6] register
			Disabled	0	Disable RR[6] register
			Enabled	1	Enable RR[6] register
H	RW	RR7			Enable or disable RR[7] register
			Disabled	0	Disable RR[7] register
			Enabled	1	Enable RR[7] register

## 40.4.7 CONFIG

Address offset: 0x50C

Configuration register

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id			C A		
Reset 0x00000001			0 1		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	SLEEP			Configure the watchdog to either be paused, or kept running, while the CPU is sleeping
			Pause	0	Pause watchdog while the CPU is sleeping
			Run	1	Keep the watchdog running while the CPU is sleeping
C	RW	HALT			Configure the watchdog to either be paused, or kept running, while the CPU is halted by the debugger
			Pause	0	Pause watchdog while the CPU is halted by the debugger
			Run	1	Keep the watchdog running while the CPU is halted by the debugger

## 40.4.8 RR[0]

Address offset: 0x600

Reload request 0

## 40.4.9 RR[1]

Address offset: 0x604

## Reload request 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	W	RR			Reload request register																											
		Reload		0x6E524635	Value to request a reload of the watchdog timer																											

## 40.4.10 RR[2]

Address offset: 0x608

## Reload request 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	W	RR			Reload request register																											
		Reload		0x6E524635	Value to request a reload of the watchdog timer																											

#### **40.4.11 RR[3]**

Address offset: 0x60C

## Reload request 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	W	RR			Reload request register																											
		Reload		0x6E524635	Value to request a reload of the watchdog timer																											

#### **40.4.12 RR[4]**

Address offset: 0x610

## Reload request 4

#### **40.4.13 RR[5]**

Address offset: 0x614

## Reload request 5

#### 40.4.14 RR[6]

Address offset: 0x618

## Reload request 6

#### 40.4.15 RR[7]

Address offset: 0x61C

## Reload request 7

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	W	RR			Reload request register																											
		Reload	0x6E524635	Value to request a reload of the watchdog timer																												

## **40.5 Electrical specification**

#### **40.5.1 Watchdog Timer Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
$I_{WDT}$	Run current for watchdog timer	0.3	2		$\mu A$
$t_{WDT}$	Time out interval		458 $\mu s$	36 h	

# 41 SWI — Software interrupts

A set of interrupts have been reserved for use as software interrupts.

## 41.1 Registers

**Table 97: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40014000	SWI	SWI0	Software interrupt 0	
0x40015000	SWI	SWI1	Software interrupt 1	
0x40016000	SWI	SWI2	Software interrupt 2	
0x40017000	SWI	SWI3	Software interrupt 3	
0x40018000	SWI	SWI4	Software interrupt 4	
0x40019000	SWI	SWI5	Software interrupt 5	

## 42 NFCT — Near field communication tag

The NFC peripheral (referred to as the 'NFC peripheral' from now on) supports communication signal interface type A and 106 kbps bit rate from the NFC Forum.

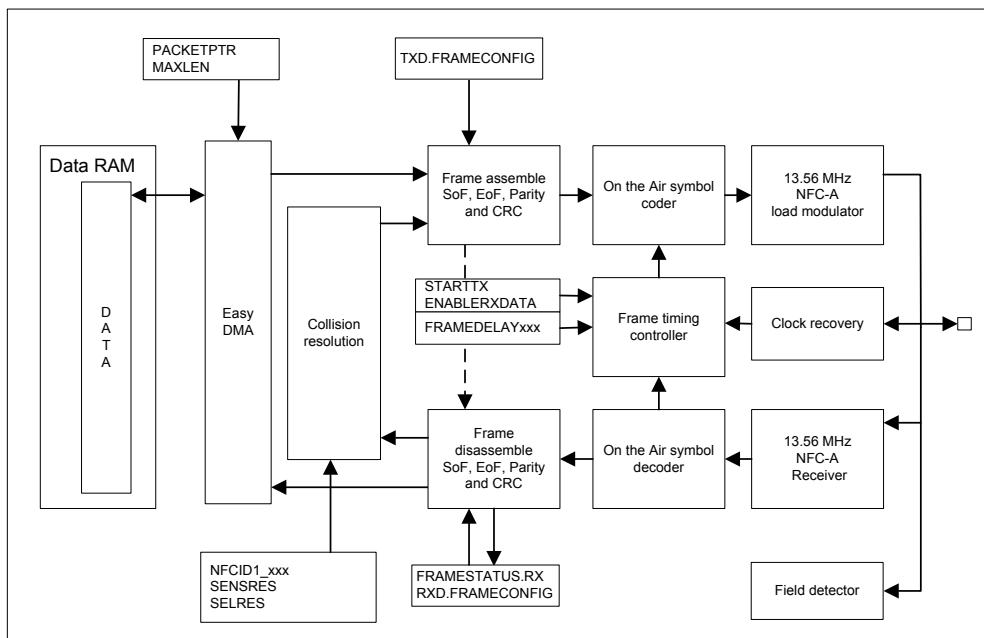
With appropriate software, the NFC peripheral can be used to emulate the listening device NFC-A as specified by the NFC Forum.

Listed here are the main features for the NFC peripheral:

- NFC-A listen mode operation
    - 13.56 MHz input frequency
    - Bit rate 106 kbps
  - Wake-on-field low power field detection (SENSE) mode
  - Frame assemble and disassemble for the NFC-A frames specified by the NFC Forum
  - Programmable frame timing controller
  - Integrated automatic collision resolution, CRC and parity functions

## 42.1 Overview

The NFC peripheral is an implementation of an NFC Forum compliant listening device NFC-A.



**Figure 117: NFC block diagram**

The NFC peripheral contains a 13.56 MHz AM receiver and a 13.56 MHz load modulator compatible with the NFC-A technology defined in the NFC Forum with 106 kbps data rate.

The received frames will be automatically disassembled and the data part of the frame transferred to RAM. When transmitting, the frame data will be transferred directly from RAM and transmitted with configurable frame type and delay timing. The system will be notified by an event whenever a complete frame is received or sent.

It also supports the collision detection and resolution ("anticollision") as defined by the NFC Forum.

Wake-on-field is supported in SENSE mode while the device is either in System OFF or System ON mode. When the antenna enters an NFC field, an event will be triggered notifying the system to activate the NFC functionality for incoming frames. In System ON, if the energy detected at the antenna increases beyond a threshold value, the module will generate a FIELDDETECTED event. The module will generate a FIELDDLOST event when the quality or strength of the field no longer support NFC communication. Please refer to [NFCT Electrical Specification](#) on page 435 for the Low Power Field Detect threshold values.

In system OFF, the NFC Low Power Field Detect function can wake the system up through a reset. The NFC bit in register [RESETREAS](#) on page 85 will be set as cause of the wake-up.

If the system is put into system OFF mode while a field is already present, the NFC Low Power Field Detect function will wake the system up right away and generate a reset.

Note that as a consequence of reset, NFC is disabled, so the reset handler will have to activate NFC again and set it up properly.

The HFXO must be running before the NFC peripheral goes into ACTIVATED state. Note that the NFC peripheral calibration is automatically done on ACTIVATE task. The HFXO can be turned off when the NFC peripheral goes into SENSE mode. The shortcut FIELDDETECTED\_ACTIVATE can be used when the HFXO is already running while in SENSE mode.

Outgoing data will be collected from RAM with the EasyDMA function and assembled according to the TXD.FRAMECONFIG register. Incoming data will be disassembled according to the RXD.FRAMECONFIG register and the data section in the frame will be written to RAM via the EasyDMA function.

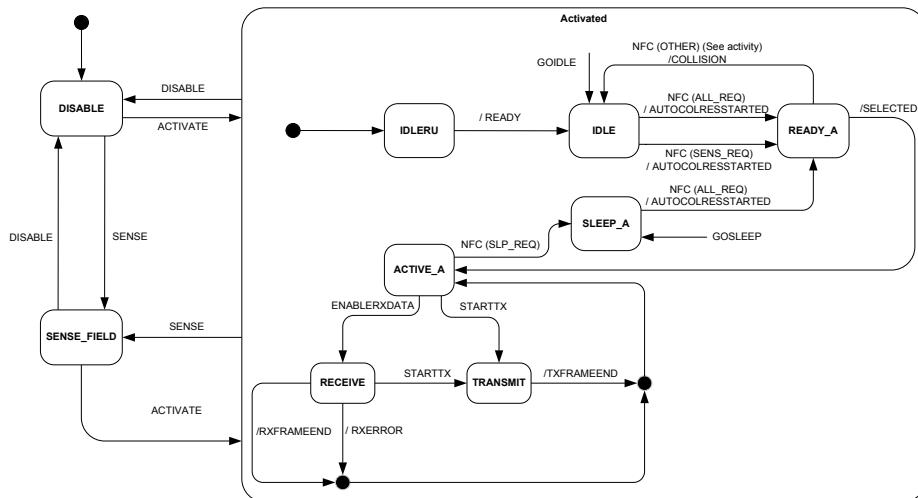
The NFC peripheral includes a frame timing controller that can be used to accurately control the inter-frame delay between the incoming frame and a corresponding outgoing frame. It also includes optional CRC functionality.

The NFC peripheral has a set of different states. The module can change state by triggering a task, or when specific operations are finalized. Events and tasks allow software to keep track of and change the current state.

See [Figure 117: NFC block diagram](#) on page 416 and [Figure 118: NFC state diagram](#) on page 418 for more information.

#### Notes:

- FIELDDLOST event will not be reflected in the state machine (for instance by going back to the DISABLE state), it is up to software to decide on the actions to take when a field lost occurs.
- FIELDDLOST event is not generated in SENSE mode.
- FIELDDETECTED event is generated only on the transition from FIELDDLOST event to energy detected by the NFC peripheral. So, sending SENSE task while field is still present does not generate FIELDDETECTED event.
- If the FIELDDETECTED event is cleared before sending the ACTIVATE task, then the FIELDDETECTED event shows up again after sending the ACTIVATE task. The shortcut FIELDDETECTED\_ACTIVATE can be used to avoid this condition.



**Figure 118: NFC state diagram**

## 42.2 Pin configuration

NFC uses two pins to connect the antenna.

These pins are shared with GPIOs, and the PROTECT field in the NFCPINS register in [UICR](#) defines the usage of these pins and their protection level against excessive voltages. The content of the NFCPINS register is reloaded at every reset.

When NFCPINS.PROTECT=NFC, a protection circuit will be enabled on the dedicated pins, preventing the chip from being damaged in the presence of a strong NFC field. The GPIO function will be disabled on those pins as well.

When NFCPINS.PROTECT=Disabled, the device will not be protected against strong NFC field damages caught by a connected NFC antenna, and the NFCT peripheral will not operate as expected, as it will never leave the DISABLE state.

The pins dedicated to the NFC antenna function will have some limitation when the pins are configured for normal GPIO operation. The pin capacitance will be higher on those (refer to  $C_{PAD\_NFC}$  in the [GPIO Electrical Specification](#) on page 154 below), and some increased leakage current between the two pins is to be expected if they are used in GPIO mode, and are driven to different logical values. To save power the two pins should always be set to the same logical value whenever entering one of the device power saving modes. Please refer to  $I_{NFC\_LEAK}$  in [GPIO Electrical Specification](#) on page 154 for details.

## 42.3 EasyDMA

The NFC peripheral implements EasyDMA for reading and writing of data packets from and to the Data RAM without CPU involvement.

The NFC EasyDMA utilizes one pointer called PACKETPTR for receiving and transmitting packets.

The EasyDMA can either read or write between the NFC peripheral and the RAM, but not at the same time. The event RXFRAMESTART indicates that the EasyDMA has started writing to the RAM for a receive frame and the event RXFRAMEND indicates that the EasyDMA has completed writing to the RAM. Similarly, the event TXFRAMESTART indicates that the EasyDMA has started reading from the RAM for a transmit frame and the event TXFRAMEND indicates that the EasyDMA has completed reading from the RAM. If a transmit and a receive operation is issued at the same time, the transmit operation would be prioritized.

Starting a transmit operation while the EasyDMA has already started writing a receive frame to the RAM will result in unpredictable behavior. Starting an EasyDMA operation whilst there is an ongoing EasyDMA operation may result in unpredictable behavior. It is recommended to wait for the TXFRAMEEND or

RXFRAKMEND event for the respective ongoing transmit or receive before starting a new receive or transmit operation.

The MAXLEN register determines the maximum number of bytes that can be read from or written to the RAM. This feature can be used to secure that the NFC peripheral does not overwrite, or read beyond, the RAM assigned to a packet. Note that if the RXD.AMOUNT or TXD.AMOUNT register indicates longer data packets than set in MAXLEN, the frames sent to or received from the physical layer will be incomplete. In RX, the OVERRUN bit in the FRAMESTATUS.RX register will be set and an RXERROR event will be triggered in that situation.

Note that RXD.AMOUNT and TXD.AMOUNT define a frame length in bytes and bits excluding SoF, EoF and parity, but including CRC for RXD.AMOUNT only, make sure to take potential additional bits into account when setting MAXLEN.

Only sending task ENABLERXDATA ensures that a new value in PACKETPTR pointing to the RX buffer in Data RAM is taken into account.

If PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Chapter [Memory](#) on page 23 for more information about the different memory regions.

The NFC peripherals normally do alternative receive and transmit frames. So, to prepare for the next frame, the PACKETPTR, MAXLEN, TXD.FRAMECONFIG and TXD.AMOUNT can be updated while the receive is in progress, and, similarly, the PACKETPTR, MAXLEN and RXD.FRAMECONFIG can be updated while the transmit is in progress. They can be updated and prepared for the next NFC frame immediately after the STARTED event of the current frame has been received. Updating the TXD.FRAMECONFIG and TXD.AMOUNT during the current transmit frame or updating RXD.FRAMECONFIG during current receive frame may cause unpredictable behaviour.

In accordance with *NFC Forum, NFC Digital Protocol Technical Specification*, the least significant bit from the least significant byte is sent on air first. The bytes are stored in increasing order, starting at the lowest address in the EasyDMA buffer in RAM.

## 42.4 Collision resolution

The NFC peripheral implements an automatic collision resolution function as defined by the NFC Forum.

The SENSRES and SELRES registers need to be programmed upfront in order for the collision resolution to behave correctly. Depending on the NFCIDSIZE field in SENSRES, the following registers also need to be programmed upfront:

- NFCID1\_LAST if NFCID1SIZE=NFCID1Single (ID = 4 bytes);
- NFCID1\_2ND\_LAST and NFCID1\_LAST if NFCID1SIZE=NFCID1Double (ID = 7 bytes);
- NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST if NFCID1SIZE=NFCID1Triple (ID = 10 bytes);

*Table 98: NFCID1 byte allocation (top sent first on air)* on page 419 explains the position of the ID bytes in NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST, depending on the ID size, and as compared to the definition used in the *NFC Forum, NFC Digital Protocol Technical Specification*.

**Table 98: NFCID1 byte allocation (top sent first on air)**

ID = 4 bytes	ID = 7 bytes	ID = 10 bytes
NFCID1_Q		nfcid1 <sub>0</sub>
NFCID1_R		nfcid1 <sub>1</sub>
NFCID1_S		nfcid1 <sub>2</sub>
NFCID1_T	nfcid1 <sub>0</sub>	nfcid1 <sub>3</sub>
NFCID1_U	nfcid1 <sub>1</sub>	nfcid1 <sub>4</sub>
NFCID1_V	nfcid1 <sub>2</sub>	nfcid1 <sub>5</sub>
NFCID1_W	nfcid1 <sub>0</sub>	nfcid1 <sub>6</sub>
NFCID1_X	nfcid1 <sub>1</sub>	nfcid1 <sub>7</sub>
NFCID1_Y	nfcid1 <sub>2</sub>	nfcid1 <sub>8</sub>
NFCID1_Z	nfcid1 <sub>3</sub>	nfcid1 <sub>9</sub>

Automatic collision resolution is enabled by default.

The hardware implementation can handle the states from IDLE to ACTIVE\_A automatically as defined in the *NFC Forum, NFC Activity Technical Specification*, and the other states are to be handled by software. The software keeps track of the state through events. The collision resolution will trigger an AUTOCOLRESSTARTED event when it has started. Reaching the ACTIVE\_A state is indicated by the SELECTED event.

If collision resolution fails, a COLLISION event is triggered. Note that errors occurring during automatic collision resolution may also cause ERROR and/or RXERROR events to be generated. Also, other events may get generated. It is recommended that the software ignores any event except COLLISION, SELECTED and FIELDLOST during automatic collision resolution. Software shall also make sure that any unwanted SHORT or PPI shortcut are disabled during automatic collision resolution.

A pre-defined set of registers, NFC.TAGHEADER0..3, containing a valid NFCID1 value, is available in [FICR](#), and can be used by software to populate the NFCID1\_3RD\_LAST, NFCID1\_2ND\_LAST and NFCID1\_LAST registers. Refer to the release notes of the NFC stack for more details on the format.

The automatic collision resolution will be restarted, if the packets are received with CRC or parity errors while in ACTIVE\_A state.

The SLP\_REQ is automatically handled by the NFC peripheral. However, this results in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) since the SLP\_REQ has no response. This error must be ignored until the SELECTED event is triggered and this error should be cleared by the software when the SELECTED event is triggered.

## 42.5 Frame timing controller

The NFC peripheral includes a frame timing controller that continuously keeps track of the number of the 13.56 MHz RF-carrier clock periods since the end of the EoF of the last received frame.

The NFC peripheral can be programmed to send a responding frame within a time window or at an exact count of RF carrier periods. In case of FRAMEDELAYMODE = Window a STARTTX task triggered before the frame timing controller counter is equal to FRAMEDELAYMIN will force the transmission to halt until the counter is equal to FRAMEDELAYMIN. If the counter is within FRAMEDELAYMIN and FRAMEDELAYMAX when the STARTTX task is triggered, the peripheral will start the transmission straight away. In case of FRAMEDELAYMODE = ExactVal, a STARTTX task, triggered before the frame delay counter is equal to FRAMEDELAYMAX, will halt the actual transmission start until the counter is equal to FRAMEDELAYMAX.

In case of FRAMEDELAYMODE = WindowGrid, the behaviour is similar to the FRAMEDELAYMODE = Window, but the actual transmission between FRAMEDELAYMIN and FRAMEDELAYMAX starts on a bit grid as defined for NFC-A Listen frames (slot duration of 128 RF carrier periods).

The FRAMEDELAYMIN and FRAMEDELAYMAX values shall only be updated before the STARTTX task is triggered. Failing to do so may cause unpredictable behaviour. An ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) will be asserted if the frame timing controller counter reaches FRAMEDELAYMAX without any STARTTX task triggered. This may happen even when the response is not required as per *NFC Forum, NFC Digital Protocol Technical Specification*. Any commands handled by the automatic collision resolution that don't involve a response being generated may also result in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS).

The frame timing controller operation is illustrated in [Figure 119: Frame timing controller \(FRAMEDELAYMODE=Window\)](#) on page 421. The frame timing controller automatically adjusts the frame timing counter based on the last received data bit according to NFC-A technology in the *NFC Forum, NFC Digital Protocol Technical Specification*.

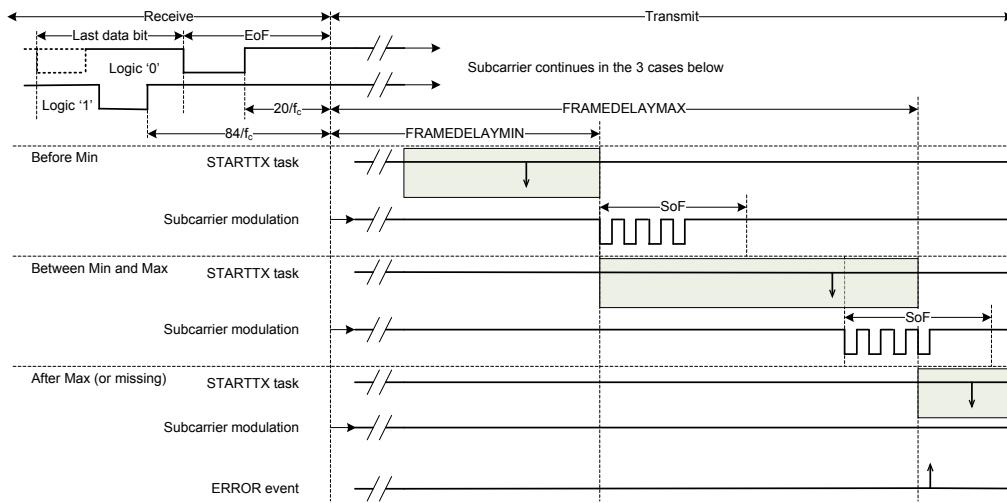


Figure 119: Frame timing controller (FRAMEDELAYMODE=Window)

## 42.6 Frame assembler

The NFC peripheral implements a frame assembler in hardware.

When the NFC peripheral is in the ACTIVE\_A state, the software can decide to enter RX or TX mode. For RX, see [Frame disassembler](#) on page 422. For TX, the software must indicate the address of the source buffer in Data RAM and its size through programming the PACKETPTR and MAXCNT registers respectively, then issuing a TXSTART task.

MAXCNT must be set so that it matches the size of the frame to be sent.

The STARTED event indicates that the PACKETPTR and MAXCNT registers have been captured by the frame assembler's EasyDMA.

When asserting the STARTTX task, the frame assembler module will start reading TXD.AMOUNT.TXDATABYTES bytes (plus one additional byte if TXD.AMOUNT.TXDATABITS > 0) from the RAM position set by the PACKETPTR.

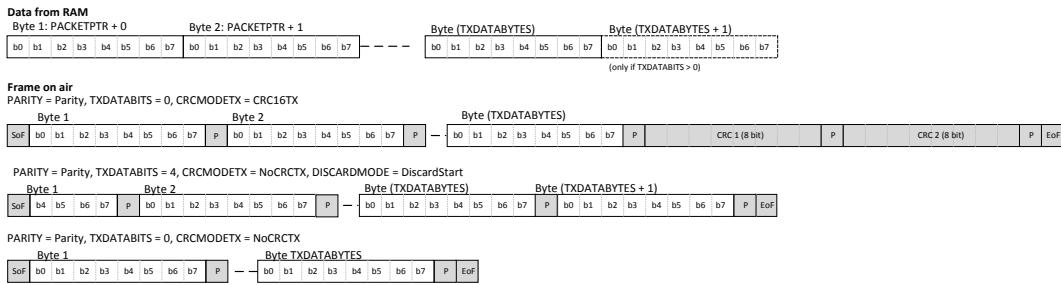
The NFC peripheral transmits the data as read from RAM, adding framing and the CRC calculated on the fly. The NFC peripheral will take (8\*TXD.AMOUNT.TXDATABYTES + TXD.AMOUNT.TXDATABITS) bits and assemble a frame according to settings in TXD.FRAMECONFIG. Both short frames, standard frames and bit oriented SDD frames as specified in the *NFC Forum, NFC Digital Protocol Technical Specification* can be assembled by correct setting of the TXD.FRAMECONFIG register.

The bytes will be transmitted on air in the same order as they are read from RAM with a rising bit order within each byte (least significant bit first). That is, b0 will be transmitted on air before b1, and so on. The bits read from RAM will be coded into symbols as defined in the *NFC Forum, NFC Digital Protocol Technical Specification*.

**Important:** Some NFC Forum documents, such as *NFC Forum, NFC Digital Protocol Technical Specification*, define bit numbering in a byte from b1 (LSB) to b8 (MSB), while most other technical documents from the NFC Forum, and also the Nordic Semiconductor documentation, traditionally numbers them from b0 to b7. The present document uses the b0 to b7 numbering scheme. Be aware of this when comparing with the *NFC Forum, NFC Digital Protocol Technical Specification* to others.

The frame assembler can be configured in TXD.FRAMECONFIG to add Start of Frame (SoF) symbol, calculate and add parity bits, and calculate and add CRC to the data read from RAM when assembling the frame. The total frame will then be longer than what is defined by TXD.AMOUNT.TXDATABYTES and TXDATABITS. DISCARDMODE will select if the first bits in the first byte read from RAM or the last bits in the last byte read from RAM will be discarded if TXD.AMOUNT.TXDATABITS are not equal to zero. Note that if TXD.FRAMECONFIG.PARITY = Parity and TXD.FRAMECONFIG.DISCARDMODE=DiscardStart, a parity bit will be included after the non-complete first byte. No parity will be added after a non-complete last byte.

The Frame Assemble operation is illustrated in [Figure 120: Frame assemble](#) on page 422 for different settings in TXD.FRAMECONFIG. All shaded bits fields are added by the frame assembler. Some of these bits are optional and appearances are configured in TXD.FRAMECONFIG. Please note that the frames illustrated do not necessarily comply with the NFC specification. The figure is only to illustrate the behavior of the NFC peripheral.



**Figure 120: Frame assemble**

The accurate timing for transmitting the frame on air is set using the frame timing controller settings.

## 42.7 Frame disassembler

The NFC peripheral implements a frame disassembler in hardware.

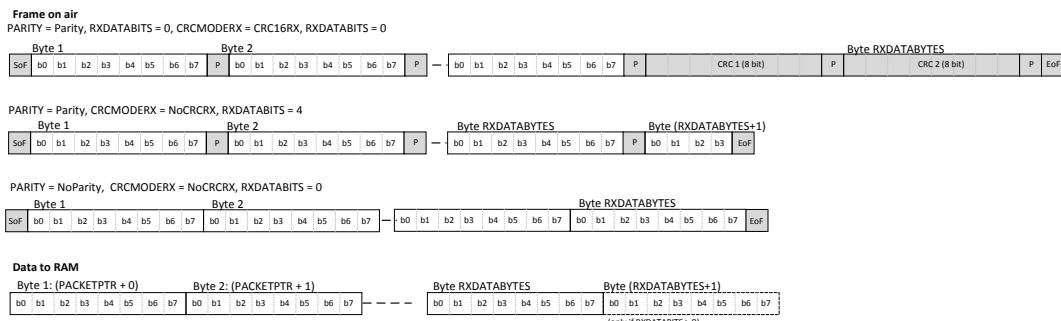
When the NFC peripheral is in the ACTIVE\_A state, the software can decide to enter RX or TX mode. For TX, see [Frame assembler](#) on page 421. For RX, the software must indicate the address of the destination buffer in Data RAM and its size through programming the PACKETPTR and MAXCNT registers respectively, then issuing a ENABLERXDATA task.

The STARTED event indicates that the PACKETPTR and MAXCNT registers have been captured by the frame disassembler's EasyDMA.

When an incoming frame starts, the RXFRAMESTART event will get issued and data will be written to the buffer in Data RAM. The frame disassembler will verify and remove on the fly any parity bits and SoF and End of Frame (EoF) symbols based on RXD.FRAMECONFIG register configuration. It will, however, verify and transfer the CRC bytes into RAM, if the CRC is was enabled through RXD.FRAMECONFIG.

When an EoF symbol is detected, the NFC peripheral will assert the RXFRAMEEND event and write the RXD.AMOUNT register to indicate numbers of received bytes and bits in the data packet. The module does not interpret the content of the data received from the remote NFC device, except for SoF, EoF, parity and CRC checking, as described above. The Frame disassemble operation is illustrated in [Figure 121: Frame disassemble illustration](#) on page 422.

Per NFC specification, the time between end of frame to the next start of frame can be as short as 86 µs, so care must be taken that PACKETPTR and MAXCNT are ready and ENABLERXDATA is issued on time after the end of previous frame. The use of a PPI shortcut from TXFRAMEEND to ENABLERXDATA is recommended.



**Figure 121: Frame disassemble illustration**

## 42.8 Antenna interface

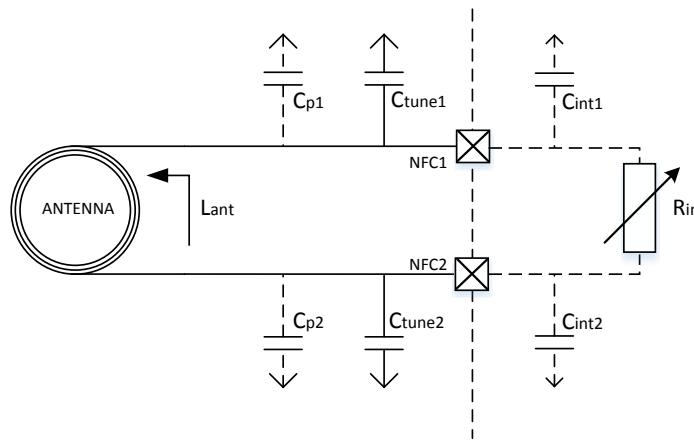
In ACTIVATED state, an amplitude regulator will adjust the voltage swing on the antenna pins to a value that is within the  $V_{swing}$  limit.

Refer to [NFCT Electrical Specification](#) on page 435.

## 42.9 NFCT antenna recommendations

The NFCT antenna coil must be connected differential between NFC1 and NFC2 pins of the device.

Two external capacitors should be used to tune the resonance of the antenna circuit to 13.56 MHz.



**Figure 122: NFCT antenna recommendations**

The required tuning capacitor value is given by the below equations:

$$C_{tune} = \frac{1}{(2\pi \cdot 13.56 \text{ MHz})^2 \cdot L_{ant}} \quad \text{where } C_{tune} = \frac{1}{2} \cdot (C_p + C_{int} + C_{tune})$$

$$\text{and } C_{tunel} = C_{tune2} = C_{tune} \quad C_{p1} = C_{p2} = C_p \quad C_{intl} = C_{int2} = C_{int}$$

$$C_{tune} = \frac{2}{(2\pi \cdot 13.56 \text{ MHz})^2 \cdot L_{ant}} - C_p - C_{int}$$

An antenna inductance of  $L_{ant} = 2 \mu\text{H}$  will give tuning capacitors in the range of 130 pF on each pin. For good performance, match the total capacitance on NFC1 and NFC2.

## 42.10 Battery protection

If the antenna is exposed to a strong NFC field, current may flow in the opposite direction on the supply due to parasitic diodes and ESD structures.

If the battery used does not tolerate return current, a series diode must be placed between the battery and the device in order to protect the battery.

## 42.11 References

NFC Forum, NFC Analog Specification version 1.0, [www.nfc-forum.org](http://www.nfc-forum.org)

NFC Forum, NFC Digital Protocol Technical Specification version 1.1, [www.nfc-forum.org](http://www.nfc-forum.org)

NFC Forum, NFC Activity Technical Specification version 1.1, [www.nfc-forum.org](http://www.nfc-forum.org)

## 42.12 Registers

**Table 99: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40005000	NFCT	NFCT	Near Field Communication Tag	

**Table 100: Register Overview**

Register	Offset	Description
TASKS_ACTIVATE	0x000	Activate NFC peripheral for incoming and outgoing frames, change state to activated
TASKS_DISABLE	0x004	Disable NFC peripheral
TASKS_SENSE	0x008	Enable NFC sense field mode, change state to sense mode
TASKS_STARTTX	0x00C	Start transmission of a outgoing frame, change state to transmit
TASKS_ENABLERXDATA	0x01C	Initializes the EasyDMA for receive.
TASKS_GOIDLE	0x024	Force state machine to IDLE state
TASKS_GOSLEEP	0x028	Force state machine to SLEEP_A state
EVENTS_READY	0x100	The NFC peripheral is ready to receive and send frames
EVENTS_FIELDDETECTED	0x104	Remote NFC field detected
EVENTS_FIELDLOST	0x108	Remote NFC field lost
EVENTS_TXFRAMESTART	0x10C	Marks the start of the first symbol of a transmitted frame
EVENTS_TXFRAMEEND	0x110	Marks the end of the last transmitted on-air symbol of a frame
EVENTS_RXFRAMESTART	0x114	Marks the end of the first symbol of a received frame
EVENTS_RXFRAMEEND	0x118	Received data have been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer
EVENTS_ERROR	0x11C	NFC error reported. The ERRORSTATUS register contains details on the source of the error.
EVENTS_RXERROR	0x128	NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.
EVENTS_ENDRX	0x12C	RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.
EVENTS_ENDTX	0x130	Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer
EVENTS_AUTOCOLRESST	0x138	Auto collision resolution process has started
EVENTS_COLLISION	0x148	NFC Auto collision resolution error reported.
EVENTS_SELECTED	0x14C	NFC Auto collision resolution successfully completed
EVENTS_STARTED	0x150	EasyDMA is ready to receive or send frames.
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSTATUS	0x404	NFC Error Status register
FRAMESTATUS.RX	0x40C	Result of last incoming frames
CURRENTLOADCTRL	0x430	Current value driven to the NFC Load Control
FIELDPRESENT	0x43C	Indicates the presence or not of a valid field
FRAMEDELAYMIN	0x504	Minimum frame delay
FRAMEDELAYMAX	0x508	Maximum frame delay
FRAMEDELAYMODE	0x50C	Configuration register for the Frame Delay Timer
PACKETPTR	0x510	Packet pointer for TXD and RXD data storage in Data RAM
MAXLEN	0x514	Size of allocated for TXD and RXD data storage buffer in Data RAM
TXD.FRAMECONFIG	0x518	Configuration of outgoing frames
TXD.AMOUNT	0x51C	Size of outgoing frame
RXD.FRAMECONFIG	0x520	Configuration of incoming frames

Register	Offset	Description
<a href="#">RXD.AMOUNT</a>	0x524	Size of last incoming frame
<a href="#">NFCID1_LAST</a>	0x590	Last NFCID1 part (4, 7 or 10 bytes ID)
<a href="#">NFCID1_2ND_LAST</a>	0x594	Second last NFCID1 part (7 or 10 bytes ID)
<a href="#">NFCID1_3RD_LAST</a>	0x598	Third last NFCID1 part (10 bytes ID)
<a href="#">SENSRES</a>	0x5A0	NFC-A SENS_RES auto-response settings
<a href="#">SELRES</a>	0x5A4	NFC-A SEL_RES auto-response settings

## 42.12.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																															B	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	FIELDDETECTED_ACTIVATE				Shortcut between FIELDDETECTED event and ACTIVATE task																										
						See <a href="#">EVENTS_FIELDDETECTED</a> and <a href="#">TASKS_ACTIVATE</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										
B	RW	FIELDLOST_SENSE				Shortcut between FIELDLOST event and SENSE task																										
						See <a href="#">EVENTS_FIELDLOST</a> and <a href="#">TASKS_SENSE</a>																										
		Disabled	0			Disable shortcut																										
		Enabled	1			Enable shortcut																										

## 42.12.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																			T	S	R	N	M	L	K	H	G	F	E	D	C	B	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																											
A	RW	READY				Enable or disable interrupt for READY event																											
						See <a href="#">EVENTS_READY</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
B	RW	FIELDDETECTED				Enable or disable interrupt for FIELDDETECTED event																											
						See <a href="#">EVENTS_FIELDDETECTED</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
C	RW	FIELDLOST				Enable or disable interrupt for FIELDLOST event																											
						See <a href="#">EVENTS_FIELDLOST</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
D	RW	TXFRAMESTART				Enable or disable interrupt for TXFRAMESTART event																											
						See <a href="#">EVENTS_TXFRAMESTART</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
E	RW	TXFRAMEEND				Enable or disable interrupt for TXFRAMEEND event																											
						See <a href="#">EVENTS_TXFRAMEEND</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
F	RW	RXFRAMESTART				Enable or disable interrupt for RXFRAMESTART event																											

### 42.12.3 INTENSET

Address offset: 0x304

## Enable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id												T	S	R		N	M	L	K		H	G	F	E	D	C	B	A				
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id		Value		Description																									
A	RW	READY	Write '1' to Enable interrupt for READY event																								See <a href="#">EVENTS_READY</a>					

				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id				T	S	R		N	M	L	K	H	G	F	E	D	C	B	A																
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																														
			Set	1	Enable																														
			Disabled	0	Read: Disabled																														
			Enabled	1	Read: Enabled																														
<b>B</b>	<b>RW</b>	<b>FIELDDETECTED</b>			Write '1' to Enable interrupt for FIELDDETECTED event																														
			Set	1	See <a href="#">EVENTS_FIELDDETECTED</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>C</b>	<b>RW</b>	<b>FIELDLOST</b>			Write '1' to Enable interrupt for FIELDLOST event																														
			Set	1	See <a href="#">EVENTS_FIELDLOST</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>D</b>	<b>RW</b>	<b>TXFRAMESTART</b>			Write '1' to Enable interrupt for TXFRAMESTART event																														
			Set	1	See <a href="#">EVENTS_TXFRAMESTART</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>E</b>	<b>RW</b>	<b>TXFRAMEEND</b>			Write '1' to Enable interrupt for TXFRAMEEND event																														
			Set	1	See <a href="#">EVENTS_TXFRAMEEND</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>F</b>	<b>RW</b>	<b>RXFRAMESTART</b>			Write '1' to Enable interrupt for RXFRAMESTART event																														
			Set	1	See <a href="#">EVENTS_RXFRAMESTART</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>H</b>	<b>RW</b>	<b>ERROR</b>			Write '1' to Enable interrupt for ERROR event																														
			Set	1	See <a href="#">EVENTS_ERROR</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>K</b>	<b>RW</b>	<b>RXERROR</b>			Write '1' to Enable interrupt for RXERROR event																														
			Set	1	See <a href="#">EVENTS_RXERROR</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>L</b>	<b>RW</b>	<b>ENDRX</b>			Write '1' to Enable interrupt for ENDRX event																														
			Set	1	See <a href="#">EVENTS_ENDRX</a>																														
			Disabled	0	Enable																														
			Enabled	1	Read: Enabled																														
<b>M</b>	<b>RW</b>	<b>ENDTX</b>			Write '1' to Enable interrupt for ENDTX event																														
			Set	1	See <a href="#">EVENTS_ENDTX</a>																														
			Disabled	0	Enable																														

#### **42.12.4 INTENCLR**

Address offset: 0x308

## Disable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
			T	S	R		N	M	L	K		H	G	F	E	D	C	B	A																																			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																							
Id																																																						
Reset 0x00000000																																																						
Id			Value Id			Value			Description																																													
A RW READY																																																						
			Clear			1			Disable																				See <a href="#">EVENTS_READY</a>																									
			Disabled			0			Read: Disabled																																													
			Enabled			1			Read: Enabled																																													
B RW FIELDDETECTED																														Write '1' to Disable interrupt for FIELDDETECTED event																								
																														See <a href="#">EVENTS_FIELDDETECTED</a>																								
			Clear			1			Disable																																													
			Disabled			0			Read: Disabled																																													
			Enabled			1			Read: Enabled																																													
C RW FIELDLOST																														Write '1' to Disable interrupt for FIELDLOST event																								
																														See <a href="#">EVENTS_FIELDLOST</a>																								
			Clear			1			Disable																																													
			Disabled			0			Read: Disabled																																													
			Enabled			1			Read: Enabled																																													
D RW TXFRAMESTART																														Write '1' to Disable interrupt for TXFRAMESTART event																								
																														See <a href="#">EVENTS_TXFRAMESTART</a>																								
			Clear			1			Disable																																													
			Disabled			0			Read: Disabled																																													
			Enabled			1			Read: Enabled																																													



Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id									T	S	R		N	M	L	K		H	G	F	E	D	C	B	A							
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
						See <a href="#">EVENTS_STARTED</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										

## 42.12.5 ERRORSTATUS

Address offset: 0x404

NFC Error Status register

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																																
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	FRAMEDELAYTIMEOUT				No STARTTX task triggered before expiration of the time set in FRAMEDELAYMAX																										
C	RW	NFCFIELDTOOstrong				Field level is too high at max load resistance																										
D	RW	NFCFIELDTOOWEAK				Field level is too low at min load resistance																										

## 42.12.6 FRAMESTATUS.RX

Address offset: 0x40C

Result of last incoming frames

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																																
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	CRCERROR				No valid End of Frame detected																										
		CRCCorrect	0			Valid CRC detected																										
		CRCError	1			CRC received does not match local check																										
B	RW	PARITYSTATUS				Parity status of received frame																										
		ParityOK	0			Frame received with parity OK																										
		ParityError	1			Frame received with parity error																										
C	RW	OVERRUN				Overrun detected																										
		NoOverrun	0			No overrun detected																										
		Overrun	1			Overrun error																										

## 42.12.7 CURRENTLOADCTRL

Address offset: 0x430

Current value driven to the NFC Load Control

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

## 42.12.8 FIELDPRESENT

Address offset: 0x43C

Indicates the presence or not of a valid field

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																															B	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value													Description															
A	R	FIELDPRESENT															Indicates the presence or not of a valid field. Available only in the activated state.															
			NoField	0													No valid field detected															
			FieldPresent	1													Valid field detected															
B	R	LOCKDETECT															Indicates if the low level has locked to the field															
			NotLocked	0													Not locked to field															
			Locked	1													Locked to field															

#### **42.12.9 FRAMEDELAYMIN**

Address offset: 0x504

## Minimum frame delay

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000480</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	FRAMEDELAYMIN				Minimum frame delay in number of 13.56 MHz clocks																										

## 42.12.10 FRAMEDELAYMAX

Address offset: 0x508

## Maximum frame delay

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00001000</b>	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	RW	FRAMEDELAYMAX			Maximum frame delay in number of 13.56 MHz clocks																											

## 42.12.11 FRAMEDELAYMODE

Address offset: 0x50C

## Configuration register for the Frame Delay Timer

## 42.12.12 PACKETPTR

Address offset: 0x510

Packet pointer for TXD and RXD data storage in Data RAM

## 42.12.13 MAXLEN

Address offset: 0x514

Size of allocated for TXD and RXD data storage buffer in Data RAM

#### **42.12.14 TXD.FRAMECONFIG**

Address offset: 0x518

## Configuration of outgoing frames

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																													D	C	B	A			
Reset 0x00000017			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
Id	RW	Field	Value Id																											Description					
A	RW	PARITY																											Adding parity or not in the frame						
																													Parity is not added in TX frames						
																													Parity is added TX frames						
B	RW	DISCARDMODE																											Discarding unused bits in start or at end of a Frame						
																													Unused bits is discarded at end of frame						
																													Unused bits is discarded at start of frame						
C	RW	SOF																											Adding SoF or not in TX frames						
																													Start of Frame symbol not added						
																													Start of Frame symbol added						
D	RW	CRCMODETX																											CRC mode for outgoing frames						
																													CRC is not added to the frame						
																													16 bit CRC added to the frame based on all the data read from RAM that is used in the frame						

#### **42.12.15 TXD.AMOUNT**

Address offset: 0x51C

## Size of outgoing frame

## 42.12.16 RXD.FRAMECONFIG

Address offset: 0x520

## Configuration of incoming frames

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																													C	B	A			
Reset 0x00000015			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	
A	RW	PARITY	Description																															
			NoParity	0	Parity expected or not in RX frame	Parity is not expected in RX frames																												
			Parity	1	Parity is expected in RX frames																													
B	RW	SOF	Description																															
			NoSoF	0	SoF expected or not in RX frames	Start of Frame symbol is not expected in RX frames																												
			SoF	1	Start of Frame symbol is expected in RX frames																													
C	RW	CRCMODERX	Description																															
			NoCRCRX	0	CRC mode for incoming frames	CRC is not expected in RX frames																												
			CRC16RX	1	Last 16 bits in RX frame is CRC, CRC is checked and CRCSTATUS updated																													

#### **42.12.17 RXD.AMOUNT**

Address offset: 0x524

## Size of last incoming frame

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																								B	B	B	B	B	B	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	RXDATABITS			Number of bits in the last byte in the frame, if less than 8 (including CRC, but excluding parity and SoF/EoF framing).																											
					Frames with 0 data bytes and less than 7 data bits are invalid and are not received properly.																											
B	R	RXDATABYTES			Number of complete bytes received in the frame (including CRC, but excluding parity and SoF/EoF framing)																											

## 42.12.18 NFCID1\_LAST

Address offset: 0x590

Last NFCID1 part (4, 7 or 10 bytes ID)

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	D	D	D	D	D	D	D	C	C	C	C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00006363</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	NFCID1_Z			NFCID1 byte Z (very last byte sent)																											
B	RW	NFCID1_Y			NFCID1 byte Y																											
C	RW	NFCID1_X			NFCID1 byte X																											
D	RW	NFCID1_W			NFCID1 byte W																											

## **42.12.19 NFCID1\_2ND\_LAST**

Address offset: 0x594

Second last NFCID1 part (7 or 10 bytes ID)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	C C C C C C C B B B B B B A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	NFCID1_V			NFCID1 byte V
B	RW	NFCID1_U			NFCID1 byte U
C	RW	NFCID1_T			NFCID1 byte T

## 42.12.20 NFCID1\_3RD\_LAST

Address offset: 0x598

Third last NFCID1 part (10 bytes ID)

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	C C C C C C C B B B B B B A A A A A A A A				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	NFCID1_S			NFCID1 byte S
B	RW	NFCID1_R			NFCID1 byte R
C	RW	NFCID1_Q			NFCID1 byte Q

## 42.12.21 SENSRES

Address offset: 0x5A0

NFC-A SENS\_RES auto-response settings

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E E E E D D D D C C B A A A A A A				
Reset 0x00000001	0 1				
Id	RW	Field	Value Id	Value	Description
A	RW	BITFRAMESDD			Bit frame SDD as defined by the b5:b1 of byte 1 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
		SDD00000	0		SDD pattern 00000
		SDD00001	1		SDD pattern 00001
		SDD00010	2		SDD pattern 00010
		SDD00100	4		SDD pattern 00100
		SDD01000	8		SDD pattern 01000
		SDD10000	16		SDD pattern 10000
B	RW	RFU5			Reserved for future use. Shall be 0.
C	RW	NFCIDSIZE			NFCID1 size. This value is used by the Auto collision resolution engine.
		NFCID1Single	0		NFCID1 size: single (4 bytes)
		NFCID1Double	1		NFCID1 size: double (7 bytes)
		NFCID1Triple	2		NFCID1 size: triple (10 bytes)
D	RW	PLATFCONFIG			Tag platform configuration as defined by the b4:b1 of byte 2 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
E	RW	RFU74			Reserved for future use. Shall be 0.

## 42.12.22 SELRES

Address offset: 0x5A4

NFC-A SEL\_RES auto-response settings

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	E D D C C B A A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	RFU10			Reserved for future use. Shall be 0.
B	RW	CASCADE	Complete	0	Cascade bit (controlled by hardware, write has no effect)
			NotComplete	1	NFCID1 complete
C	RW	RFU43			Reserved for future use. Shall be 0.
D	RW	PROTOCOL			Protocol as defined by the b7:b6 of SEL_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
E	RW	RFU7			Reserved for future use. Shall be 0.

## 42.13 Electrical specification

### 42.13.1 NFCT Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$f_c$	Frequency of operation		13.56		MHz
$C_{MI}$	Carrier modulation index	95			%
DR	Data Rate	106			kbps
$f_s$	Modulation sub-carrier frequency		$f_c/16$		MHz
$V_{swing}$	Peak differential Input voltage swing on NFC1 and NFC2		VDD		Vp
$V_{sense}$	Peak differential Field detect threshold level on NFC1-NFC2 <sup>36</sup>	1.0			Vp
$I_{sense}$	Current in SENSE STATE	100			nA
$I_{activated}$	Current in ACTIVATED STATE	480			µA
$R_{in\_min}$	Minimum input resistance when regulating voltage swing		40		Ω
$R_{in\_max}$	Maximum input resistance when regulating voltage swing	1.0			kΩ
$R_{in\_loadmod}$	Input resistance when load modulating	8	22		Ω
$I_{max}$	Maximum input current on NFC pins		80		mA

### 42.13.2 NFCT Timing Parameters

Symbol	Description	Min.	Typ.	Max.	Units
$t_{activate}$	Time from task_ACTIVATE in SENSE or DISABLE state to ACTIVATE_A or IDLE state <sup>37</sup>		500		us
$t_{sense}$	Time from remote field is present in SENSE mode to FIELDDETECTED event is asserted	20			us

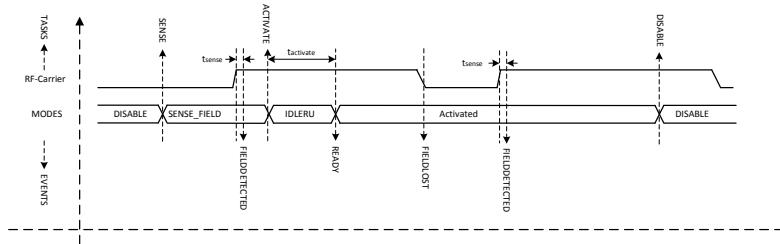


Figure 123: NFCT timing parameters (Shortcuts for FIELDDETECTED and FIELDLOST are disabled)

<sup>36</sup> Input is high impedance in sense mode

<sup>37</sup> Does not account for voltage supply and oscillator startup times

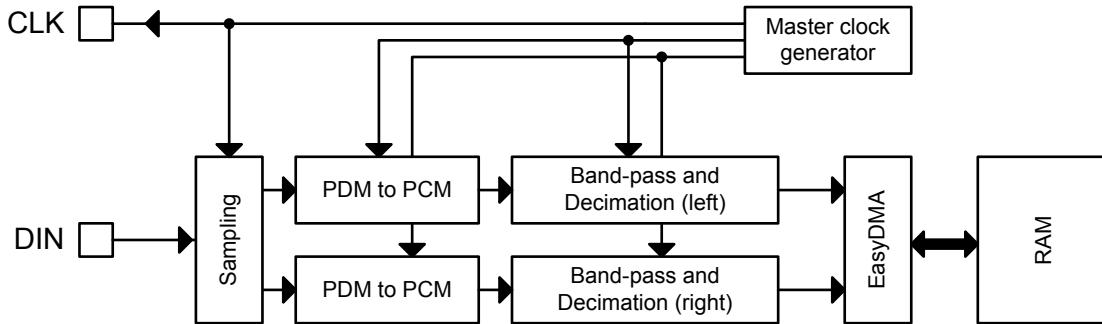
## 43 PDM — Pulse density modulation interface

The pulse density modulation (PDM) module enables input of pulse density modulated signals from external audio frontends, for example, digital microphones. The PDM module generates the PDM clock and supports single-channel or dual-channel (Left and Right) data input. Data is transferred directly to RAM buffers using EasyDMA.

Listed here are the main features for PDM:

- Up to two PDM microphones configured as a Left/Right pair using the same data input
- 16 kHz output sample rate, 16-bit samples
- EasyDMA support for sample buffering
- HW decimation filters

The PDM module illustrated in [Figure 124: PDM module](#) on page 436 is interfacing up to two digital microphones with the PDM interface. It implements EasyDMA, which relieves real-time requirements associated with controlling the PDM slave from a low priority CPU execution context. It also includes all the necessary digital filter elements to produce PCM samples. The PDM module allows continuous audio streaming.



**Figure 124: PDM module**

### 43.1 Master clock generator

The FREQ field in the master clock's PDMCLKCTRL register allows adjusting the PDM clock's frequency.

The master clock generator does not add any jitter to the HFCLK source chosen. It is recommended (but not mandatory) to use the Xtal as HFCLK source.

### 43.2 Module operation

By default, bits from the left PDM microphone are sampled on PDM\_CLK falling edge, bits for the right are sampled on the rising edge of PDM\_CLK, resulting in two bitstreams. Each bitstream is fed into a digital filter which converts the PDM stream into 16-bit PCM samples, and filters and down-samples them to reach the appropriate sample rate.

The EDGE field in the MODE register allows swapping Left and Right, so that Left will be sampled on rising edge, and Right on falling.

The PDM module uses EasyDMA to store the samples coming out from the filters into one buffer in RAM.

Depending on the mode chosen in the OPERATION field in the MODE register, memory either contains alternating left and right 16-bit samples (Stereo), or only left 16-bit samples (Mono).

To ensure continuous PDM sampling, it is up to the application to update the EasyDMA destination address pointer as the previous buffer is filled.

The continuous transfer can be started or stopped by sending the START and STOP tasks. STOP becomes effective after the current frame has finished transferring, which will generate the STOPPED event. The STOPPED event indicates that all activity in the module are finished, and that the data is available in RAM (EasyDMA has finished transferring as well). Attempting to restart before receiving the STOPPED event may result in unpredictable behaviour.

### 43.3 Decimation filter

In order to convert the incoming data stream into PCM audio samples, a decimation filter is included in the PDM interface module.

The input of the filter is the two-channel PDM serial stream (with left channel on clock high, right channel on clock low), its output is  $2 \times 16$ -bit PCM samples at a sample rate 64 times lower than the PDM clock rate.

The filter stage of each channel is followed by a digital volume control, to attenuate or amplify the output samples in a range of -20 dB to +20 dB around the default (reset) setting, defined by  $G_{PDM,\text{default}}$ . The gain is controlled by the GAINL and GAINR registers.

As an example, if the goal is to achieve 2500 RMS output samples (16 bit) with a 1 kHz 90 dBA signal into a -26 dBFS sensitivity PDM microphone, the user will have to sum the PDM module's default gain ( $G_{PDM,\text{default}}$ ) and the gain introduced by the microphone and acoustic path of his implementation (an attenuation would translate into a negative gain), and adjust GAINL and GAINR by this amount. Assuming that only the PDM module influences the gain, GAINL and GAINR must be set to  $-G_{PDM,\text{default}}$  dB to achieve the requirement.

With  $G_{PDM,\text{default}}=3.2$  dB, and as GAINL and GAINR are expressed in 0.5 dB steps, the closest value to program would be 3.0 dB, which can be calculated as:

```
GAINL = GAINR = (DefaultGain - (2 * 3))
```

Remember to check that the resulting values programmed into GAINL and GAINR fall within MinGain and MaxGain.

### 43.4 EasyDMA

Samples will be written directly to RAM, and EasyDMA must be configured accordingly.

The address pointer for the EasyDMA channel is set in SAMPLE.PTR register. If the destination address set in SAMPLE.PTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

DMA supports Stereo (Left+Right 16-bit samples) and Mono (Left only) data transfer, depending on setting in the OPERATION field in the MODE register. The samples are stored little endian.

**Table 101: DMA sample storage**

MODE.OPERATION	Bits per sample	Result stored per RAM word	Physical RAM allocated (32 bit words)	Result boundary indexes in RAM	Note
Stereo	32 (2x16)	L+R	ceil(SAMPLE.MAXCNT/2)	R0=[31:16]; L0=[15:0]	Default
Mono	16	2xL	ceil(SAMPLE.MAXCNT/2)	L1=[31:16]; L0=[15:0]	

The destination buffer in RAM consists of one block, the size of which is set in SAMPLE.MAXCNT register. Format is number of 16-bit samples. The physical RAM allocated is always:

```
(RAM allocation, in bytes) = SAMPLE.MAXCNT * 2;
```

(but the mapping of the samples depends on MODE.OPERATION).

If OPERATION=Stereo, RAM will contain a succession of Left and Right samples.

If OPERATION=Mono, RAM will contain a succession of mono samples.

For a given value of SAMPLE.MAXCNT, the buffer in RAM can contain half the stereo sampling time as compared to the mono sampling time.

The PDM acquisition can be started by the START task, after the SAMPLE.PTR and SAMPLE.MAXCNT registers have been written. When starting the module, it will take some time for the filters to start outputting valid data. Transients from the PDM microphone itself may also occur. The first few samples (typically around 50) might hence contain invalid values or transients. It is therefore advised to discard the first few samples after a PDM start.

As soon as the STARTED event is received, the firmware can write the next SAMPLE.PTR value (this register is double-buffered), to ensure continuous operation.

When the buffer in RAM is filled with samples, an END event is triggered. The firmware can start processing the data in the buffer. Meanwhile, the PDM module starts acquiring data into the new buffer pointed to by SAMPLE.PTR, and sends a new STARTED event, so that the firmware can update SAMPLE.PTR to the next buffer address.

## 43.5 Hardware example

Connect the microphone clock to CLK, and data to DIN.

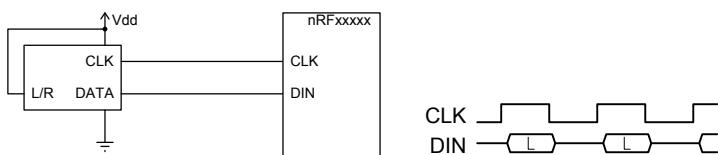


Figure 125: Example of a single PDM microphone, wired as left

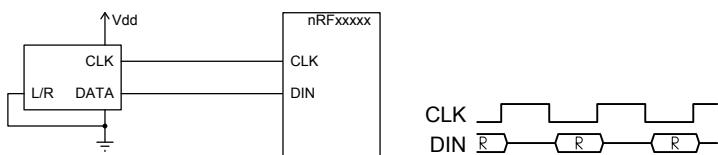


Figure 126: Example of a single PDM microphone, wired as right

Note that in a single-microphone (mono) configuration, depending on the microphone's implementation, either the left or the right channel (sampled at falling or rising CLK edge respectively) will contain reliable data. If two microphones are used, one of them has to be set as left, the other as right (L/R pin tied high or to GND on the respective microphone). It is strongly recommended to use two microphones of exactly the same brand and type so that their timings in left and right operation match.

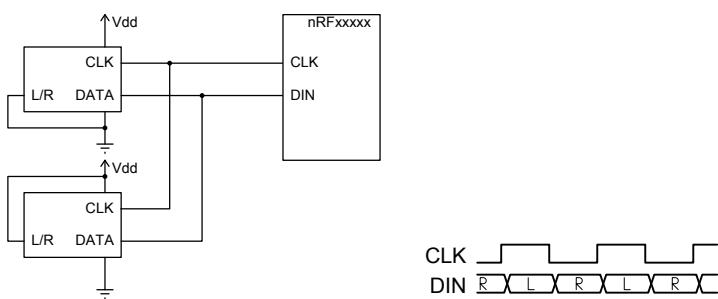


Figure 127: Example of two PDM microphones

## 43.6 Pin configuration

The CLK and DIN signals associated to the PDM module are mapped to physical pins according to the configuration specified in the PSEL.CLK and PSEL.DIN registers respectively. If the CONNECT field in any PSEL register is set to Disconnected, the associated PDM module signal will not be connected to the required physical pins, and will not operate properly.

The PSEL.CLK and PSEL.DIN registers and their configurations are only used as long as the PDM module is enabled, and retained only as long as the device is in System ON mode. See [POWER — Power supply](#) on page 78 for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN\_CNF[n] register.

To ensure correct behaviour in the PDM module, the pins used by the PDM module must be configured in the GPIO peripheral as described in [Table 102: GPIO configuration before enabling peripheral](#) on page 439 before enabling the PDM module. This is to ensure that the pins used by the PDM module are driven correctly if the PDM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the PDM module is supposed to be connected to an external PDM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behaviour.

**Table 102: GPIO configuration before enabling peripheral**

PDM signal	PDM pin	Direction	Output value	Comment
CLK	As specified in PSEL.CLK	Output	0	
DIN	As specified in PSEL.DIN	Input	Not applicable	

## 43.7 Registers

**Table 103: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4001D000	PDM	PDM	Pulse Density Modulation (Digital Microphone Interface)	

**Table 104: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Starts continuous PDM transfer
TASKS_STOP	0x004	Stops PDM transfer
EVENTS_STARTED	0x100	PDM transfer has started
EVENTS_STOPPED	0x104	PDM transfer has finished
EVENTS_END	0x108	The PDM has written the last sample specified by SAMPLE.MAXCNT (or the last sample after a STOP task has been received) to Data RAM
<a href="#">INTEN</a>	0x300	Enable or disable interrupt
<a href="#">INTENSET</a>	0x304	Enable interrupt
<a href="#">INTENCLR</a>	0x308	Disable interrupt
<a href="#">ENABLE</a>	0x500	PDM module enable register
<a href="#">PDMCLKCTRL</a>	0x504	PDM clock generator control
<a href="#">MODE</a>	0x508	Defines the routing of the connected PDM microphones' signals
<a href="#">GAINL</a>	0x518	Left output gain adjustment
<a href="#">GAINR</a>	0x51C	Right output gain adjustment
<a href="#">PSEL.CLK</a>	0x540	Pin number configuration for PDM CLK signal
<a href="#">PSEL.DIN</a>	0x544	Pin number configuration for PDM DIN signal
<a href="#">SAMPLE.PTR</a>	0x560	RAM address pointer to write samples to with EasyDMA
<a href="#">SAMPLE.MAXCNT</a>	0x564	Number of samples to allocate memory for in EasyDMA mode

### 43.7.1 INTEN

Address offset: 0x300

Enable or disable interrupt

## 43.7.2 INTENSET

Address offset: 0x304

## Enable interrupt

### 43.7.3 INTENCLR

Address offset: 0x308

Disable interrupt

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

#### **43.7.4 ENABLE**

Address offset: 0x500

## PDM module enable register

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

## 43.7.5 PDMCLKCTRL

Address offset: 0x504

## PDM clock generator control

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x08400000</b>	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	FREQ			PDM_CLK frequency																											
		1000K	0x08000000	PDM_CLK = 32 MHz / 32 = 1.000 MHz																												
		Default	0x08400000	PDM_CLK = 32 MHz / 31 = 1.032 MHz																												
		1067K	0x08800000	PDM_CLK = 32 MHz / 30 = 1.067 MHz																												

## 43.7.6 MODE

Address offset: 0x508

Defines the routing of the connected PDM microphones' signals

### **43.7.7 GAINL**

Address offset: 0x518

## Left output gain adjustment

43.7.8 GAINR

Address offset: 0x51C

### Right output gain adjustment

### 43.7.9 PSEL.CLK

Address offset: 0x540

### Pin number configuration for PDM CLK signal

## **43.7.10 PSEL.DIN**

Address offset: 0x544

## Pin number configuration for PDM DIN signal

### **43.7.11 SAMPLE.PTR**

Address offset: 0x560

RAM address pointer to write samples to with EasyDMA

### **43.7.12 SAMPLE.MAXCNT**

Address offset: 0x564

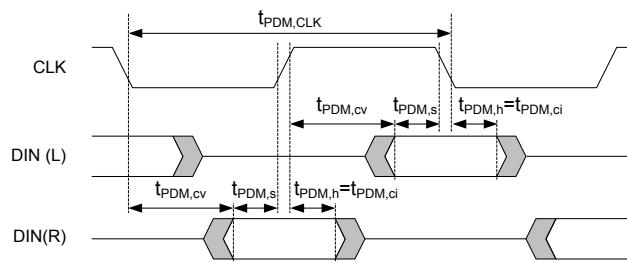
Number of samples to allocate memory for in EasyDMA mode

## **43.8 Electrical specification**

### **43.8.1 PDM Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
$I_{PDM, stereo}$	PDM module active current, stereo operation <sup>38</sup>		1.4		mA
$f_{PDM, CLK}$	PDM clock speed		1.032		MHz
$t_{PDM, JITTER}$	Jitter in PDM clock output			20	ns
$T_{dPDM, CLK}$	PDM clock duty cycle	40	50	60	%
$t_{PDM, DATA}$	Decimation filter delay			5	ms
$t_{PDM, cv}$	Allowed clock edge to data valid			125	ns
$t_{PDM, ci}$	Allowed (other) clock edge to data invalid	0			ns
$t_{PDM, s}$	Data setup time at $f_{PDM, CLK}=1.024$ MHz	65			ns
$t_{PDM, h}$	Data hold time at $f_{PDM, CLK}=1.024$ MHz	0			ns
$G_{PDM, default}$	Default (reset) absolute gain of the PDM module		3.2		dB

<sup>38</sup> Average current including PDM and DMA transfers, excluding clock and power supply base currents



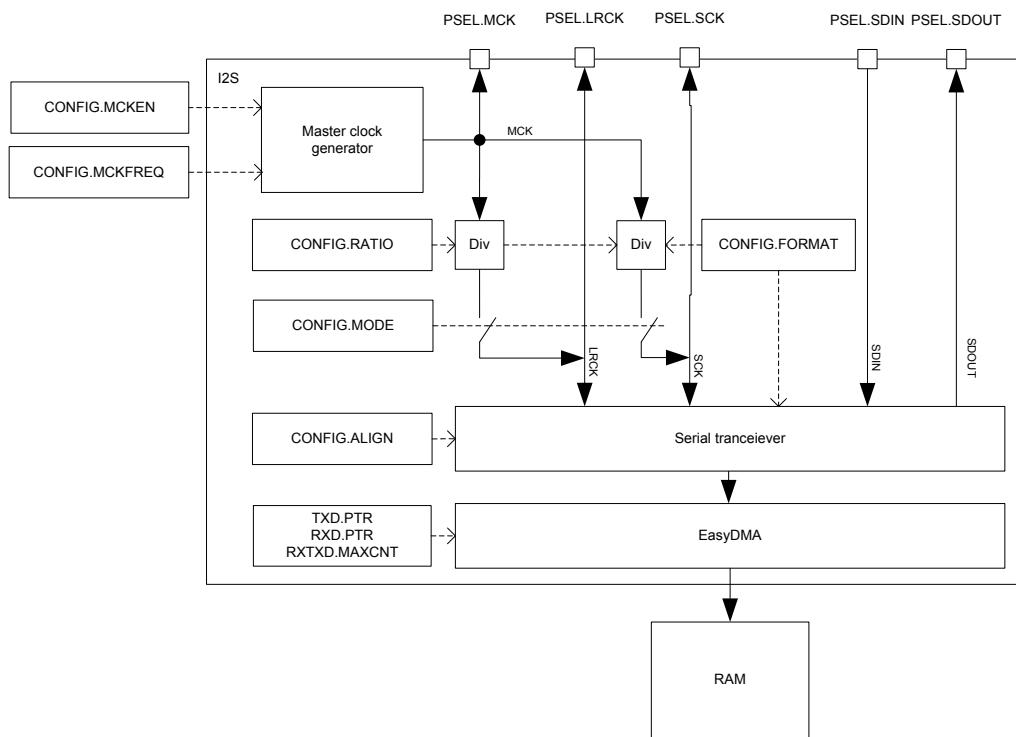
**Figure 128: PDM timing diagram**

## 44 I<sup>2</sup>S — Inter-IC sound interface

The I<sup>2</sup>S (Inter-IC Sound) module, supports the original two-channel I<sup>2</sup>S format, and left or right-aligned formats. It implements EasyDMA for sample transfer directly to and from RAM without CPU intervention.

The I<sup>2</sup>S peripheral has the following main features:

- Master and Slave mode
- Simultaneous bi-directional (TX and RX) audio streaming
- Original I<sup>2</sup>S and left- or right-aligned format
- 8, 16 and 24-bit sample width
- Low-jitter Master Clock generator
- Various sample rates



**Figure 129: I<sup>2</sup>S master**

### 44.1 Mode

The I<sup>2</sup>S protocol specification defines two modes of operation, Master and Slave.

The I<sup>2</sup>S mode decides which of the two sides (Master or Slave) shall provide the clock signals LRCK and SCK, and these signals are always supplied by the Master to the Slave.

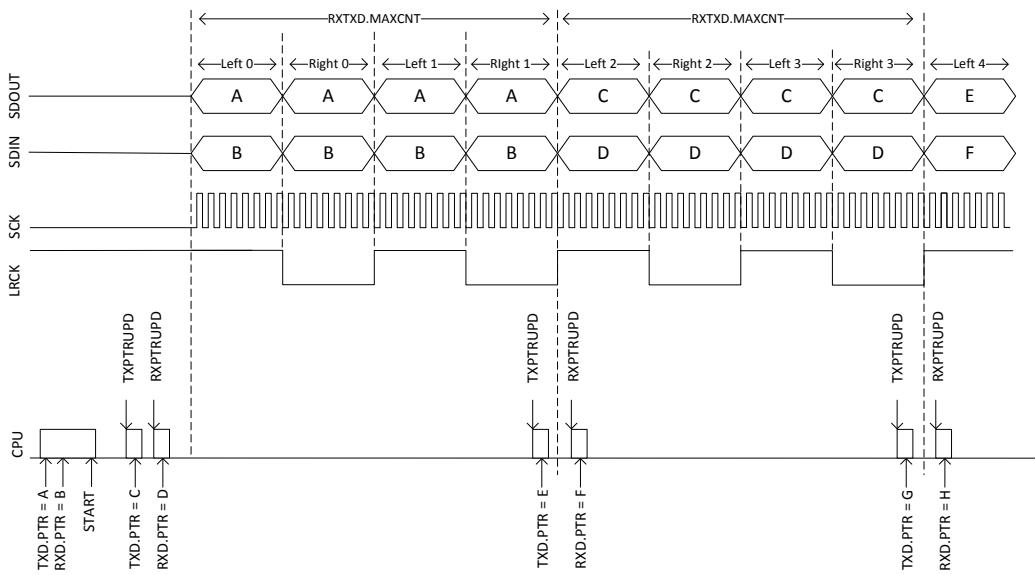
### 44.2 Transmitting and receiving

The I<sup>2</sup>S module supports both transmission (TX) and reception (RX) of serial data. In both cases the serial data is shifted synchronously to the clock signals SCK and LRCK.

TX data is written to the SDOUT pin on the falling edge of SCK, and RX data is read from the SDIN pin on the rising edge of SCK. The most significant bit (MSB) is always transmitted first.

TX and RX are available in both Master and Slave modes and can be enabled/disabled independently in the [CONFIG.TXEN](#) on page 455 and [CONFIG.RXEN](#) on page 455.

Transmission and/or reception is started by triggering the START task. When started and transmission is enabled (in [CONFIG.TXEN](#) on page 455), the TXPTRUPD event will be generated for every [RXTXD.MAXCNT](#) on page 458 number of transmitted data words (containing one or more samples). Similarly, when started and reception is enabled (in [CONFIG.RXEN](#) on page 455), the RXPTRUPD event will be generated for every [RXTXD.MAXCNT](#) on page 458 received data words.



**Figure 130: Transmitting and receiving. CONFIG.FORMAT = Aligned, CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo, RXTXD.MAXCNT = 1.**

### 44.3 Left right clock (LRCK)

The Left Right Clock (LRCK), often referred to as "word clock", "sample clock" or "word select" in I<sup>2</sup>S context, is the clock defining the frames in the serial bit streams sent and received on SDOUT and SDIN, respectively.

In I<sup>2</sup>S mode, each frame contains one left and right sample pair, with the left sample being transferred during the low half period of LRCK followed by the right sample being transferred during the high period of LRCK.

In Aligned mode, each frame contains one left and right sample pair, with the left sample being transferred during the high half period of LRCK followed by the right sample being transferred during the low period of LRCK.

Consequently, the LRCK frequency is equivalent to the audio sample rate.

When operating in Master mode, the LRCK is generated from the MCK, and the frequency of LRCK is then given as:

$$\text{LRCK} = \text{MCK} / \text{CONFIG.RATIO}$$

LRCK always toggles around the falling edge of the serial clock SCK.

### 44.4 Serial clock (SCK)

The serial clock (SCK), often referred to as the serial bit clock, pulses once for each data bit being transferred on the serial data lines SDIN and SDOUT.

When operating in Master mode the SCK is generated from the MCK, and the frequency of SCK is then given as:

$$SCK = 2 * LRCK * CONFIG.SWIDTH$$

The falling edge of the SCK falls on the toggling edge of LRCK.

When operating in Slave mode SCK is provided by the external I<sup>2</sup>S master.

## 44.5 Master clock (MCK)

The master clock (MCK) is the clock from which LRCK and SCK are derived when operating in Master mode.

The MCK is generated by an internal MCK generator. This generator always needs to be enabled when in Master mode, but the generator can also be enabled when in Slave mode. Enabling the generator when in slave mode can be useful in the case where the external Master is not able to generate its own master clock.

The MCK generator is enabled/disabled in the register [CONFIG.MCKEN](#) on page 456, and the generator is started or stopped by the START or STOP tasks.

In Master mode the LRCK and the SCK frequencies are closely related, as both are derived from MCK and set indirectly through [CONFIG.RATIO](#) on page 456 and [CONFIG.SWIDTH](#) on page 457.

When configuring these registers, the user is responsible for fulfilling the following requirements:

1. SCK frequency can never exceed the MCK frequency, which can be formulated as:

$$CONFIG.RATIO \geq 2 * CONFIG.SWIDTH$$

2. The MCK/LRCK ratio shall be a multiple of  $2 * CONFIG.SWIDTH$ , which can be formulated as:

$$\text{Integer} = (CONFIG.RATIO / (2 * CONFIG.SWIDTH))$$

The MCK signal can be routed to an output pin (specified in PSEL.MCK) to supply external I<sup>2</sup>S devices that require the MCK to be supplied from the outside.

When operating in Slave mode, the I<sup>2</sup>S module does not use the MCK and the MCK generator does not need to be enabled.

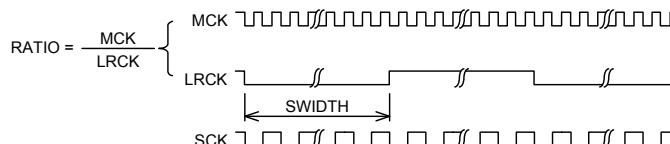


Figure 131: Relation between RATIO, MCK and LRCK.

Table 105: Configuration examples

Desired LRCK [Hz]	CONFIG.SWIDTH	CONFIG.RATIO	CONFIG.MCKFREQ	MCK [Hz]	LRCK [Hz]	LRCK error [%]
16000	16Bit	32X	32MDIV63	507936.5	15873.0	-0.8
16000	16Bit	64X	32MDIV31	1032258.1	16129.0	0.8
16000	16Bit	256X	32MDIV8	4000000.0	15625.0	-2.3
32000	16Bit	32X	32MDIV31	1032258.1	32258.1	0.8
32000	16Bit	64X	32MDIV16	2000000.0	31250.0	-2.3
32000	16Bit	256X	32MDIV4	8000000.0	31250.0	-2.3
44100	16Bit	32X	32MDIV23	1391304.3	43478.3	-1.4
44100	16Bit	64X	32MDIV11	2909090.9	45454.5	3.1
44100	16Bit	256X	32MDIV3	10666666.7	41666.7	-5.5

## 44.6 Width, alignment and format

The CONFIG.SWIDTH register primarily defines the sample width of the data written to memory. In master mode, it then also sets the amount of bits per frame. In Slave mode it controls padding/trimming if required. Left, right, transmitted, and received samples always have the same width. The CONFIG FORMAT register specifies the position of the data frames with respect to the LRCK edges in both Master and Slave modes.

When using I<sup>2</sup>S format, the first bit in a half-frame (containing one left or right sample) gets sampled on the second rising edge of the SCK after a LRCK edge. When using Aligned mode, the first bit in a half-frame gets sampled on the first rising edge of SCK following a LRCK edge.

For data being received on SDIN the sample value can be either right or left-aligned inside a half-frame, as specified in [CONFIG.ALIGN](#) on page 457. [CONFIG.ALIGN](#) on page 457 affects only the decoding of the incoming samples (SDIN), while the outgoing samples (SDOUT) are always left-aligned (or justified).

When using left-alignment, each half-frame starts with the MSB of the sample value (both for data being sent on SDOUT and received on SDIN).

When using right-alignment, each half-frame of data being received on SDIN ends with the LSB of the sample value, while each half-frame of data being sent on SDOUT starts with the MSB of the sample value (same as for left-alignment).

In Master mode, the size of a half-frame (in number of SCK periods) equals the sample width (in number of bits), and in this case the alignment setting does not care as each half-frame in any case will start with the MSB and end with the LSB of the sample value.

In slave mode, however, the sample width does not need to equal the frame size. This means you might have extra or fewer SCK pulses per half-frame than what the sample width specified in [CONFIG.SWIDTH](#) requires.

In the case where we use **left-alignment** and the number of SCK pulses per half-frame is **higher** than the sample width, the following will apply:

- For data received on SDIN, all bits after the LSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0.

In the case where we use **left-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

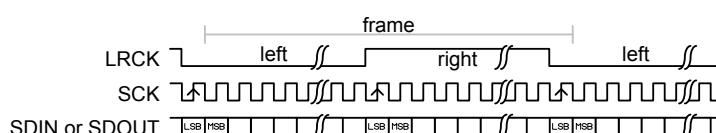
- Data sent and received on SDOUT and SDIN will be truncated with the LSBs being removed first.

In the case where we use **right-alignment** and the number of SCK pulses per frame is **higher** than the sample width, the following will apply:

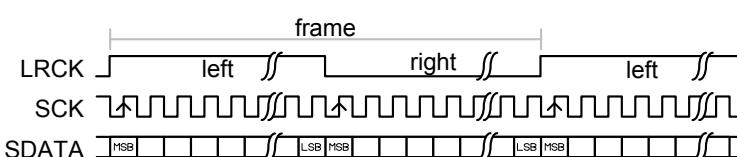
- For data received on SDIN, all bits before the MSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0 (same behavior as for left-alignment).

In the case where we use **right-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

- Data received on SDIN will be sign-extended to "sample width" number of bits before being written to memory.
- Data sent on SDOUT will be truncated with the LSBs being removed first (same behavior as for left-alignment).



**Figure 132: I<sup>2</sup>S format. CONFIG.SWIDTH equalling half-frame size.**



**Figure 133: Aligned format. CONFIG.SWIDTH equalling half-frame size.**

## 44.7 EasyDMA

The I<sup>2</sup>S module implements EasyDMA for accessing internal Data RAM without CPU intervention.

The source and destination pointers for the TX and RX data are configured in [TXD.PTR](#) on page 458 and [RXD.PTR](#) on page 458. The memory pointed to by these pointers will only be read or written when TX or RX are enabled in [CONFIG.TXEN](#) on page 455 and [CONFIG.RXEN](#) on page 455.

The addresses written to the pointer registers [TXD.PTR](#) on page 458 and [RXD.PTR](#) on page 458 are double-buffered in hardware, and these double buffers are updated for every [RXTXD.MAXCNT](#) on page 458 words (containing one or more samples) read/written from/to memory. The events TXPTRUPD and RXPTRUPD are generated whenever the TXD.PTR and RXD.PTR are transferred to these double buffers.

If [TXD.PTR](#) on page 458 is not pointing to the Data RAM region when transmission is enabled, or [RXD.PTR](#) on page 458 is not pointing to the Data RAM region when reception is enabled, an EasyDMA transfer may result in a HardFault and/or memory corruption. See [Memory](#) on page 23 for more information about the different memory regions.

Due to the nature of I<sup>2</sup>S, where the number of transmitted samples always equals the number of received samples (at least when both TX and RX are enabled), one common register [RXTXD.MAXCNT](#) on page 458 is used for specifying the sizes of these two memory buffers. The size of the buffers is specified in a number of 32-bit words. Such a 32-bit memory word can either contain four 8-bit samples, two 16-bit samples or one right-aligned 24-bit sample sign extended to 32 bit.

In stereo mode (decided by CONFIG.CHANNELS), the samples are stored as "left and right sample pairs" in memory. Figure [Figure 134: Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.](#), [Figure 136: Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.](#) on page 449, [Figure 138: Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.](#) on page 450 and [Figure 135: Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.](#) on page 450, [Figure 137: Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.](#) on page 450 and [Figure 139: Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.](#) on page 451 show how the samples are mapped to memory in this mode.

In mono mode (left or right channel only), samples for only one single channel are stored in memory. Figure [Figure 135: Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.](#) on page 450, [Figure 137: Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.](#) on page 450 and [Figure 139: Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.](#) on page 451 show how samples are mapped to memory in this mode.

	31	24 23	16 15	8 7	0
x.PTR		Right sample 1	Left sample 1	Right sample 0	Left sample 0
x.PTR + 4		Right sample 3	Left sample 3	Right sample 2	Left sample 2
x.PTR + (n*2) - 4		Left sample n-1	Right sample n-1	Right sample n-2	Left sample n-2

**Figure 134: Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.**

	31	24 23	16 15	8 7	0
x.PTR		Left sample 3	Left sample 2	Left sample 1	Left sample 0
x.PTR + 4		Left sample 7	Left sample 6	Left sample 5	Left sample 4
x.PTR + n - 4		Left sample n-1	Left sample n-2	Left sample n-3	Left sample n-4

Figure 135: Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.

	31	16 15	0
x.PTR		Right sample 0	Left sample 0
x.PTR + 4		Right sample 1	Left sample 1
x.PTR + (n*4) - 4		Right sample n - 1	Left sample n - 1

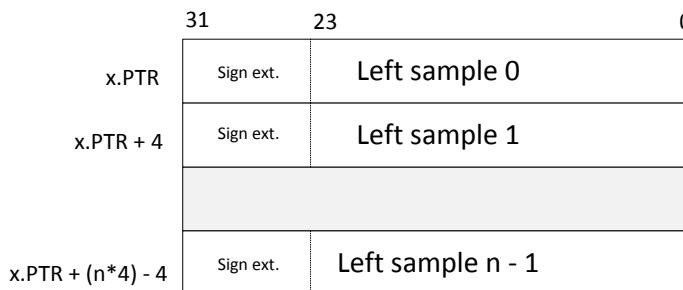
Figure 136: Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.

	31	16 15	0
x.PTR		Left sample 1	Left sample 0
x.PTR + 4		Left sample 3	Left sample 2
x.PTR + (n*2) - 4		Left sample n - 1	Left sample n - 2

Figure 137: Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.

	31	23	0
x.PTR		Sign ext.	Left sample 0
x.PTR + 4		Sign ext.	Right sample 0
x.PTR + (n*8) - 8		Sign ext.	Left sample n - 1
x.PTR + (n*8) - 4		Sign ext.	Right sample n - 1

Figure 138: Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.



**Figure 139: Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.**

## 44.8 Module operation

Described here is a typical operating procedure for the I<sup>2</sup>S module.

### 1. Configure the I<sup>2</sup>S module using the CONFIG registers

```
// Enable reception
NRF_I2S->CONFIG.RXEN = (I2S_CONFIG_RXEN_RXEN_Enabled <<
                           I2S_CONFIG_RXEN_RXEN_Pos);
// Enable transmission
NRF_I2S->CONFIG.TXEN = (I2S_CONFIG_TXEN_TXEN_Enabled <<
                           I2S_CONFIG_TXEN_TXEN_Pos);
// Enable MCK generator
NRF_I2S->CONFIG.MCKEN = (I2S_CONFIG_MCKEN_MCKEN_Enabled <<
                           I2S_CONFIG_MCKEN_MCKEN_Pos);
// MCKFREQ = 4 MHz
NRF_I2S->CONFIG.MCKFREQ = I2S_CONFIG_MCKFREQ_MCKFREQ_32MDIV8 <<
                           I2S_CONFIG_MCKFREQ_MCKFREQ_Pos;
// Ratio = 256
NRF_I2S->CONFIG.RATIO = I2S_CONFIG_RATIO_RATIO_256X <<
                           I2S_CONFIG_RATIO_RATIO_Pos;
// MCKFREQ = 4 MHz and Ratio = 256 gives sample rate = 15.625 ks/s
// Sample width = 16 bit
NRF_I2S->CONFIG.SWIDTH = I2S_CONFIG_SWIDTH_SWIDTH_16Bit <<
                           I2S_CONFIG_SWIDTH_SWIDTH_Pos;
// Alignment = Left
NRF_I2S->CONFIG.ALIGN = I2S_CONFIG_ALIGN_ALIGN_Left <<
                           I2S_CONFIG_ALIGN_ALIGN_Pos;
// Format = I2S
NRF_I2S->CONFIG.FORMAT = I2S_CONFIG_FORMAT_FORMAT_I2S <<
                           I2S_CONFIG_FORMAT_FORMAT_Pos;
// Use stereo
NRF_I2S->CONFIG.CHANNELS = I2S_CONFIG_CHANNELS_CHANNELS_Stereo <<
                           I2S_CONFIG_CHANNELS_CHANNELS_Pos;
```

### 2. Map IO pins using the PINSEL registers

```
// MCK routed to pin 0
NRF_I2S->PSEL.MCK = (0 << I2S_PSEL_MCK_CONNECT_Pos) |
                           (I2S_PSEL_MCK_CONNECT_Connected <<
                           I2S_PSEL_MCK_CONNECT_Pos);
// SCK routed to pin 1
NRF_I2S->PSEL.SCK = (1 << I2S_PSEL_SCK_CONNECT_Pos) |
                           (I2S_PSEL_SCK_CONNECT_Connected <<
                           I2S_PSEL_SCK_CONNECT_Pos);
// LRCK routed to pin 2
NRF_I2S->PSEL.LRCK = (2 << I2S_PSEL_LRCK_CONNECT_Pos) |
                           (I2S_PSEL_LRCK_CONNECT_Connected <<
```

```

I2S_PSEL_LRCK_CONNECT_Pos);
// SDOUT routed to pin 3
NRF_I2S->PSEL.SDOUT = (3 << I2S_PSEL_SDOUT_CONNECT_Pos) |
(I2S_PSEL_SDOUT_CONNECT_Connected <<

I2S_PSEL_SDOUT_CONNECT_Pos);
// SDIN routed on pin 4
NRF_I2S->PSEL.SDIN = (4 << I2S_PSEL_SDIN_CONNECT_Pos) |
(I2S_PSEL_SDIN_CONNECT_Connected <<

I2S_PSEL_SDIN_CONNECT_Pos);

```

### 3. Configure TX and RX data pointers using the TXD, RXD and RXTXD registers

```

NRF_I2S->TXD.PTR = my_tx_buf;
NRF_I2S->RXD.PTR = my_rx_buf;
NRF_I2S->TXD.MAXCNT = MY_BUF_SIZE;

```

### 4. Enable the I<sup>2</sup>S module using the ENABLE register

```
NRF_I2S->ENABLE = 1;
```

### 5. Start audio streaming using the START task

```
NRF_I2S->TASKS_START = 1;
```

### 6. Handle received and transmitted data when receiving the TXPTRUPD and RXPTRUPD events

```

if (NRF_I2S->EVENTS_TXPTRUPD != 0)
{
    NRF_I2S->TXD.PTR = my_next_tx_buf;
    NRF_I2S->EVENTS_TXPTRUPD = 0;
}

if (NRF_I2S->EVENTS_RXPTRUPD != 0)
{
    NRF_I2S->RXD.PTR = my_next_rx_buf;
    NRF_I2S->EVENTS_RXPTRUPD = 0;
}

```

## 44.9 Pin configuration

The MCK, SCK, LRCK, SDIN and SDOUT signals associated with the I<sup>2</sup>S module are mapped to physical pins according to the pin numbers specified in the PSEL.x registers.

These pins are acquired whenever the I<sup>2</sup>S module is enabled through the register [ENABLE](#) on page 455.

When a pin is acquired by the I<sup>2</sup>S module, the direction of the pin (input or output) will be configured automatically, and any pin direction setting done in the GPIO module will be overridden. The directions for the various I<sup>2</sup>S pins are shown below in [Table 106: GPIO configuration before enabling peripheral \(master mode\)](#) on page 452 and [Table 107: GPIO configuration before enabling peripheral \(slave mode\)](#) on page 453.

To secure correct signal levels on the pins when the system is in OFF mode, and when the I<sup>2</sup>S module is disabled, these pins must be configured in the GPIO peripheral directly.

**Table 106: GPIO configuration before enabling peripheral (master mode)**

I <sup>2</sup> S signal	I <sup>2</sup> S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Output	0	

I <sup>2</sup> S signal	I <sup>2</sup> S pin	Direction	Output value	Comment
SCK	As specified in PSEL.SCK	Output	0	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

**Table 107: GPIO configuration before enabling peripheral (slave mode)**

I <sup>2</sup> S signal	I <sup>2</sup> S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

## 44.10 Registers

**Table 108: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40025000	I2S	I2S	Inter-IC Sound Interface	

**Table 109: Register Overview**

Register	Offset	Description
TASKS_START	0x000	Starts continuous I2S transfer. Also starts MCK generator when this is enabled.
TASKS_STOP	0x004	Stops I2S transfer. Also stops MCK generator. Triggering this task will cause the {event:STOPPED} event to be generated.
EVENTS_RXPTRUPD	0x104	The RXD.PTR register has been copied to internal double-buffers. When the I2S module is started and RX is enabled, this event will be generated for every RXTXD.MAXCNT words that are received on the SDIN pin.
EVENTS_STOPPED	0x108	I2S transfer stopped.
EVENTS_TXPTRUPD	0x114	The TDX.PTR register has been copied to internal double-buffers. When the I2S module is started and TX is enabled, this event will be generated for every RXTXD.MAXCNT words that are sent on the SDOUT pin.
<b>INTEN</b>	0x300	Enable or disable interrupt
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>ENABLE</b>	0x500	Enable I2S module.
<b>CONFIG.MODE</b>	0x504	I2S mode.
<b>CONFIG.RXEN</b>	0x508	Reception (RX) enable.
<b>CONFIG.TXEN</b>	0x50C	Transmission (TX) enable.
<b>CONFIG.MCKEN</b>	0x510	Master clock generator enable.
<b>CONFIG.MCKFREQ</b>	0x514	Master clock generator frequency.
<b>CONFIG.RATIO</b>	0x518	MCK / LRCK ratio.
<b>CONFIG.SWIDTH</b>	0x51C	Sample width.
<b>CONFIG.ALIGN</b>	0x520	Alignment of sample within a frame.
<b>CONFIG FORMAT</b>	0x524	Frame format.
<b>CONFIG CHANNELS</b>	0x528	Enable channels.
<b>RXD.PTR</b>	0x538	Receive buffer RAM start address.
<b>TXD.PTR</b>	0x540	Transmit buffer RAM start address.
<b>RXTXD.MAXCNT</b>	0x550	Size of RXD and TXD buffers.
<b>PSEL.MCK</b>	0x560	Pin select for MCK signal.
<b>PSEL.SCK</b>	0x564	Pin select for SCK signal.
<b>PSEL.LRCK</b>	0x568	Pin select for LRCK signal.
<b>PSEL.SDIN</b>	0x56C	Pin select for SDIN signal.
<b>PSEL.SDOUT</b>	0x570	Pin select for SDOUT signal.

### 44.10.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F C B				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
B	RW	RXPTRUPD			Enable or disable interrupt for RXPTRUPD event See <a href="#">EVENTS_RXPTRUPD</a>
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	STOPPED			Enable or disable interrupt for STOPPED event See <a href="#">EVENTS_STOPPED</a>
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXPTRUPD			Enable or disable interrupt for TXPTRUPD event See <a href="#">EVENTS_TXPTRUPD</a>
			Disabled	0	Disable
			Enabled	1	Enable

## 44.10.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F C B				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
B	RW	RXPTRUPD			Write '1' to Enable interrupt for RXPTRUPD event See <a href="#">EVENTS_RXPTRUPD</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to Enable interrupt for STOPPED event See <a href="#">EVENTS_STOPPED</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXPTRUPD			Write '1' to Enable interrupt for TXPTRUPD event See <a href="#">EVENTS_TXPTRUPD</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

## 44.10.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F C B				
Reset 0x00000000	0 0				
Id	RW	Field	Value Id	Value	Description
B	RW	RXPTRUPD			Write '1' to Disable interrupt for RXPTRUPD event See <a href="#">EVENTS_RXPTRUPD</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to Disable interrupt for STOPPED event

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F C B				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_STOPPED</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
F RW TXPTRUPD					Write '1' to Disable interrupt for TXPTRUPD event
					See <a href="#">EVENTS_TXPTRUPD</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

#### 44.10.4 ENABLE

Address offset: 0x500

Enable I2S module.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A RW ENABLE					Enable I2S module.
		Disabled	0	Disable	
		Enabled	1	Enable	

#### 44.10.5 CONFIG.MODE

Address offset: 0x504

I2S mode.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A RW MODE					I2S mode.
		Master	0	Master mode. SCK and LRCK generated from internal master clcok (MCK) and output on pins defined by PSEL.xxx.	
		Slave	1	Slave mode. SCK and LRCK generated by external master and received on pins defined by PSEL.xxx	

#### 44.10.6 CONFIG.RXEN

Address offset: 0x508

Reception (RX) enable.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A RW RXEN					Reception (RX) enable.
		Disabled	0	Reception disabled and now data will be written to the RXD.PTR address.	
		Enabled	1	Reception enabled.	

#### 44.10.7 CONFIG.TXEN

Address offset: 0x50C

Transmission (TX) enable.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TXEN			Transmission (TX) enable.																											
		Disabled	0		Transmission disabled and now data will be read from the RXD.TXD address.																											
		Enabled	1		Transmission enabled.																											

## 44.10.8 CONFIG.MCKEN

Address offset: 0x510

Master clock generator enable.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	MCKEN			Master clock generator enable.																											
			Disabled	0	Master clock generator disabled and PSEL.MCK not connected(available as GPIO).																											
			Enabled	1	Master clock generator running and MCK output on PSEL.MCK.																											

#### **44.10.9 CONFIG.MCKFREQ**

Address offset: 0x514

Master clock generator frequency.

## **44.10.10 CONFIG.RATIO**

Address offset: 0x518

## MCK / LRCK ratio.

#### **44.10.11 CONFIG.SWIDTH**

Address offset: 0x51C

Sample width.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A																														
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	SWIDTH				Sample width.																										
		8Bit		0		8 bit.																										
		16Bit		1		16 bit.																										
		24Bit		2		24 bit.																										

## **44.10.12 CONFIG.ALIGN**

Address offset: 0x520

## Alignment of sample within a frame.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	ALIGN			Alignment of sample within a frame.																											
		Left	0	Left-aligned.																												
		Right	1	Right-aligned.																												

#### **44.10.13 CONFIG FORMAT**

Address offset: 0x524

## Frame format.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																										
A	RW	FORMAT				Frame format.																										
		I2S		0		Original I2S format.																										
		Aligned		1		Alternate (left- or right-aligned) format.																										

#### **44.10.14 CONFIG.CHANNELS**

Address offset: 0x528

## Enable channels.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																															A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value			Description																									
A	RW	CHANNELS					Enable channels.																									
		Stereo		0			Stereo.																									
		Left		1			Left only.																									
		Right		2			Right only.																									

## **44.10.15 RXD.PTR**

Address offset: 0x538

Receive buffer RAM start address.

44.10.16 TXD.PTR

Address offset: 0x540

Transmit buffer RAM start address.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	PTR			Transmit buffer Data RAM start address. When transmitting, words containing samples will be fetched from this address. This address is a word aligned Data RAM address.																											

## 44.10.17 RXTXD.MAXCNT

Address offset: 0x550

Size of RXD and TXD buffers.

44.10.18 PSEL.MCK

Address offset: 0x560

Pin select for MCK signal.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

44.10.19 PSEL.SCK

Address offset: 0x564

Pin select for SCK signal.

44.10.20 PSEL.LRCK

Address offset: 0x568

Pin select for LRCK signal.

44.10.21 PSEL,SDIN

Address offset: 0x56C

Pin select for SDIN signal.

44.10.22 PSEL.SDOUT

Address offset: 0x570

Pin select for SDOUT signal.

## 44.11 Electrical specification

### 44.11.1 I2S timing specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{S\_SDIN}$	SDIN setup time before SCK rising	20			ns
$t_{H\_SDIN}$	SDIN hold time after SCK rising	15			ns
$t_{S\_SDOUT}$	SDOUT setup time after SCK falling	40			ns
$t_{H\_SDOUT}$	SDOUT hold time before SCK falling	6			ns
$t_{SCK\_LRCK}$	SCLK falling to LRCK edge	-5	0	5	ns
$f_{MCK}$	MCK frequency			4000	kHz
$f_{LRCK}$	LRCK frequency			48	kHz
$f_{SCK}$	SCK frequency			2000	kHz
$DC_{CK}$	Clock duty cycle (MCK, LRCK, SCK)	45		55	%

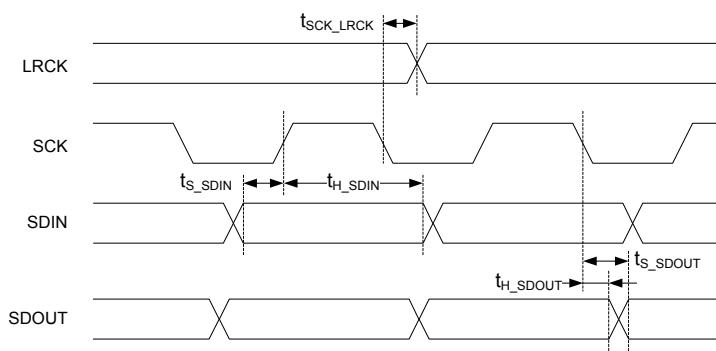


Figure 140: I<sup>2</sup>S timing diagram

## 45 MWU — Memory watch unit

The Memory watch unit (MWU) can be used to generate events when a memory region is accessed by the CPU. The MWU can be configured to trigger events for access to Data RAM and Peripheral memory segments. The MWU allows an application developer to generate memory access events during development for debugging or during production execution for failure detection and recovery.

Listed here are the main features for MWU:

- Six memory regions, four user-configurable and two fixed regions in peripheral address space
- Flexible configuration of regions with START and END addresses
- Generate events on CPU read and/or write to a defined region of Data RAM or peripheral memory address space
- Programmable maskable or non-maskable (NMI) interrupt on events
- Peripheral interfaces can be watched for read and write access using subregions of the two fixed memory regions

**Table 110: Memory regions**

Memory region	START address	END address
REGION[0..3]	Configurable	Configurable
PREGION[0]	0x40000000	0x4001FFFF
PREGION[1]	0x40020000	0x4003FFFF

Each MWU region is defined by a start address and an end address, configured by the START and END registers respectively. These addresses are byte aligned and inclusive. The END register value has to be greater or equal to the START register value. Each region is associated with a pair of events that indicate that either a write access or a read access from the CPU has been detected inside the region.

For regions containing subregions (see below), a set of status registers PERREGION[0..1].SUBSTATWA and PERREGION[0..1].SUBSTATRA indicate which subregion(s) caused the EVENT\_PREGION[0..1].WA and EVENT\_PREGION[0..1].RA respectively.

The MWU is only able to detect memory accesses in the Data RAM and Peripheral memory segments from the CPU, see [Memory](#) on page 23 for more information about the different memory segments. EasyDMA accesses are not monitored by the MWU. The MWU requires two HCLK cycles to detect and generate the event.

The peripheral regions, PREGION[0..1], are divided into 32 equally sized subregions, SR[0..31]. All subregions are excluded in the main region by default, and any can be included by specifying them in the SUBS register. When a subregion is excluded from the main region, the memory watch mechanism will not trigger any events when that subregion is accessed.

Subregions in PREGION[0..1] cannot be individually configured for read or write access watch. Watch configuration is only possible for a region as a whole. The PRGNiRA and PRGNiWA ( $i=0..1$ ) fields in the REGIONEN register control watching read and write access.

REGION[0..3] can be individually enabled for read and/or write access watching through their respective RGNIWA and RGNIWA ( $i=0..3$ ) fields in the REGIONEN register.

REGIONENSET and REGIONENCLR allow respectively enabling and disabling one or multiple REGIONS or PREGIONS watching in a single write access.

### 45.1 Registers

**Table 111: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40020000	MWU	MWU	Memory Watch Unit	

**Table 112: Register Overview**

Register	Offset	Description
EVENTS_REGION[0].WA	0x100	Write access to region 0 detected
EVENTS_REGION[0].RA	0x104	Read access to region 0 detected
EVENTS_REGION[1].WA	0x108	Write access to region 1 detected
EVENTS_REGION[1].RA	0x10C	Read access to region 1 detected
EVENTS_REGION[2].WA	0x110	Write access to region 2 detected
EVENTS_REGION[2].RA	0x114	Read access to region 2 detected
EVENTS_REGION[3].WA	0x118	Write access to region 3 detected
EVENTS_REGION[3].RA	0x11C	Read access to region 3 detected
EVENTS_PREGION[0].WA	0x160	Write access to peripheral region 0 detected
EVENTS_PREGION[0].RA	0x164	Read access to peripheral region 0 detected
EVENTS_PREGION[1].WA	0x168	Write access to peripheral region 1 detected
EVENTS_PREGION[1].RA	0x16C	Read access to peripheral region 1 detected
<b>INTEN</b>	0x300	Enable or disable interrupt
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>NMIEN</b>	0x320	Enable or disable non-maskable interrupt
<b>NMIENSET</b>	0x324	Enable non-maskable interrupt
<b>NMIENCLR</b>	0x328	Disable non-maskable interrupt
<b>PERREGION[0].SUBSTATI</b>	0x400	Source of event/interrupt in region 0, write access detected while corresponding subregion was enabled for watching
<b>PERREGION[0].SUBSTATI</b>	0x404	Source of event/interrupt in region 0, read access detected while corresponding subregion was enabled for watching
<b>PERREGION[1].SUBSTATI</b>	0x408	Source of event/interrupt in region 1, write access detected while corresponding subregion was enabled for watching
<b>PERREGION[1].SUBSTATI</b>	0x40C	Source of event/interrupt in region 1, read access detected while corresponding subregion was enabled for watching
<b>REGIONEN</b>	0x510	Enable/disable regions watch
<b>REGIONENSET</b>	0x514	Enable regions watch
<b>REGIONENCLR</b>	0x518	Disable regions watch
<b>REGION[0].START</b>	0x600	Start address for region 0
<b>REGION[0].END</b>	0x604	End address of region 0
<b>REGION[1].START</b>	0x610	Start address for region 1
<b>REGION[1].END</b>	0x614	End address of region 1
<b>REGION[2].START</b>	0x620	Start address for region 2
<b>REGION[2].END</b>	0x624	End address of region 2
<b>REGION[3].START</b>	0x630	Start address for region 3
<b>REGION[3].END</b>	0x634	End address of region 3
<b>PREGION[0].START</b>	0x6C0	Reserved for future use
<b>PREGION[0].END</b>	0x6C4	Reserved for future use
<b>PREGION[0].SUBS</b>	0x6C8	Subregions of region 0
<b>PREGION[1].START</b>	0x6D0	Reserved for future use
<b>PREGION[1].END</b>	0x6D4	Reserved for future use
<b>PREGION[1].SUBS</b>	0x6D8	Subregions of region 1

### 45.1.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	L	K	J	I	H	G	F	E	D	C	B																					
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value	Id	Value	Description																										
A	RW	REGIONOWA				Enable or disable interrupt for REGION[0].WA event																										
						See <a href="#">EVENTS_REGION[0].WA</a>																										
						Disabled																										
						Disable																										

## 45.1.2 INTENSET

Address offset: 0x304

Enable interrupt



Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	L K J I H G F E D C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_PREGION[1].WA</a>
L	RW	PREGION1RA	Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
					Write '1' to Enable interrupt for PREGION[1].RA event
					See <a href="#">EVENTS_PREGION[1].RA</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

### 45.1.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	L K J I H G F E D C B A				
<b>Reset 0x00000000</b>	0 0				
Id	RW	Field	Value Id	Value	Description
A	RW	REGION0WA	Set	1	Write '1' to Disable interrupt for REGION[0].WA event
			Disabled	0	See <a href="#">EVENTS_REGION[0].WA</a>
			Enabled	1	Disable
B	RW	REGION0RA	Set	1	Read: Disabled
			Disabled	0	Read: Enabled
			Enabled	1	Write '1' to Disable interrupt for REGION[0].RA event
					See <a href="#">EVENTS_REGION[0].RA</a>
C	RW	REGION1WA	Set	1	Write '1' to Disable interrupt for REGION[1].WA event
			Disabled	0	See <a href="#">EVENTS_REGION[1].WA</a>
			Enabled	1	Disable
D	RW	REGION1RA	Set	1	Read: Disabled
			Disabled	0	Read: Enabled
			Enabled	1	Write '1' to Disable interrupt for REGION[1].RA event
					See <a href="#">EVENTS_REGION[1].RA</a>
E	RW	REGION2WA	Set	1	Write '1' to Disable interrupt for REGION[2].WA event
			Disabled	0	See <a href="#">EVENTS_REGION[2].WA</a>
			Enabled	1	Disable
F	RW	REGION2RA	Set	1	Read: Disabled
			Disabled	0	Read: Enabled
			Enabled	1	Write '1' to Disable interrupt for REGION[2].RA event
					See <a href="#">EVENTS_REGION[2].RA</a>
G	RW	REGION3WA	Set	1	Write '1' to Disable interrupt for REGION[3].WA event
			Disabled	0	See <a href="#">EVENTS_REGION[3].WA</a>
			Enabled	1	Disable

## **45.1.4 NMIEN**

Address offset: 0x320

Enable or disable non-maskable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
			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		
A	RW	REGION0WA	Value Id		Value		Description																												
							Enable or disable non-maskable interrupt for REGION[0].WA event																												
					Disabled		0		See <a href="#">EVENTS_REGION[0].WA</a>																										
B	RW	REGION0RA			Enabled		1		Disable																										
					Disabled		0		Enable																										
					Enabled		1		See <a href="#">EVENTS_REGION[0].RA</a>																										
C	RW	REGION1WA					Disabled		0		Enable or disable non-maskable interrupt for REGION[1].WA event																								
					Enabled		1		See <a href="#">EVENTS_REGION[1].WA</a>																										
					Disabled		0		Disable																										

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id		L K J I H G F E D C B A			
<b>Reset 0x00000000</b>		<b>0 0</b>			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
D	RW	REGION1RA			Enable or disable non-maskable interrupt for REGION[1].RA event See <a href="#">EVENTS_REGION[1].RA</a>
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	REGION2WA			Enable or disable non-maskable interrupt for REGION[2].WA event See <a href="#">EVENTS_REGION[2].WA</a>
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	REGION2RA			Enable or disable non-maskable interrupt for REGION[2].RA event See <a href="#">EVENTS_REGION[2].RA</a>
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	REGION3WA			Enable or disable non-maskable interrupt for REGION[3].WA event See <a href="#">EVENTS_REGION[3].WA</a>
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	REGION3RA			Enable or disable non-maskable interrupt for REGION[3].RA event See <a href="#">EVENTS_REGION[3].RA</a>
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	PREGION0WA			Enable or disable non-maskable interrupt for PREGION[0].WA event See <a href="#">EVENTS_PREGION[0].WA</a>
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	PREGION0RA			Enable or disable non-maskable interrupt for PREGION[0].RA event See <a href="#">EVENTS_PREGION[0].RA</a>
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	PREGION1WA			Enable or disable non-maskable interrupt for PREGION[1].WA event See <a href="#">EVENTS_PREGION[1].WA</a>
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	PREGION1RA			Enable or disable non-maskable interrupt for PREGION[1].RA event See <a href="#">EVENTS_PREGION[1].RA</a>
			Disabled	0	Disable
			Enabled	1	Enable

## 45.1.5 NMIENSET

Address offset: 0x324

Enable non-maskable interrupt



Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id			L K J I	H G F E D C B A	
Reset 0x00000000			0 0		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	PREGION0RA			Write '1' to Enable non-maskable interrupt for PREGION[0].RA event See <a href="#">EVENTS_PREGION[0].RA</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	PREGION1WA			Write '1' to Enable non-maskable interrupt for PREGION[1].WA event See <a href="#">EVENTS_PREGION[1].WA</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	PREGION1RA			Write '1' to Enable non-maskable interrupt for PREGION[1].RA event See <a href="#">EVENTS_PREGION[1].RA</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

#### 45.1.6 NMIEENCLR

Address offset: 0x328

Disable non-maskable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Id			L K J I	H G F E D C B A	
Reset 0x00000000			0 0		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	REGION0WA			Write '1' to Disable non-maskable interrupt for REGION[0].WA event See <a href="#">EVENTS_REGION[0].WA</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REGION0RA			Write '1' to Disable non-maskable interrupt for REGION[0].RA event See <a href="#">EVENTS_REGION[0].RA</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	REGION1WA			Write '1' to Disable non-maskable interrupt for REGION[1].WA event See <a href="#">EVENTS_REGION[1].WA</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	REGION1RA			Write '1' to Disable non-maskable interrupt for REGION[1].RA event See <a href="#">EVENTS_REGION[1].RA</a>

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			L	K	J	I																												
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Id	RW	Field	Value	Id	Value	Description																												
E RW REGION2WA			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
F RW REGION2RA			Write '1' to Disable non-maskable interrupt for REGION[2].WA event																															
			Clear	1	Disable	See <a href="#">EVENTS_REGION[2].WA</a>																												
			Disabled	0	Read: Disabled																													
G RW REGION3WA			Write '1' to Disable non-maskable interrupt for REGION[3].WA event																															
			Clear	1	Disable	See <a href="#">EVENTS_REGION[3].WA</a>																												
			Disabled	0	Read: Disabled																													
H RW REGION3RA			Write '1' to Disable non-maskable interrupt for REGION[3].RA event																															
			Clear	1	Disable	See <a href="#">EVENTS_REGION[3].RA</a>																												
			Disabled	0	Read: Disabled																													
I RW PREGION0WA			Write '1' to Disable non-maskable interrupt for PREGION[0].WA event																															
			Clear	1	Disable	See <a href="#">EVENTS_PREGION[0].WA</a>																												
			Disabled	0	Read: Disabled																													
J RW PREGION0RA			Write '1' to Disable non-maskable interrupt for PREGION[0].RA event																															
			Clear	1	Disable	See <a href="#">EVENTS_PREGION[0].RA</a>																												
			Disabled	0	Read: Disabled																													
K RW PREGION1WA			Write '1' to Disable non-maskable interrupt for PREGION[1].WA event																															
			Clear	1	Disable	See <a href="#">EVENTS_PREGION[1].WA</a>																												
			Disabled	0	Read: Disabled																													
L RW PREGION1RA			Write '1' to Disable non-maskable interrupt for PREGION[1].RA event																															
			Clear	1	Disable	See <a href="#">EVENTS_PREGION[1].RA</a>																												
			Disabled	0	Read: Disabled																													

#### **45.1.7 PERREGION[0].SUBSTATWA**

Address offset: 0x400

Source of event/interrupt in region 0, write access detected while corresponding subregion was enabled for watching

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
<b>Id RW Field</b>			<b>Value Id</b>	<b>Value</b>	<b>Description</b>																													
			NoAccess	0	No write access occurred in this subregion																													
			Access	1	Write access(es) occurred in this subregion																													
R RW SR17			NoAccess	0	Subregion 17 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
S RW SR18			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 18 in region 0 (write '1' to clear)																													
T RW SR19			NoAccess	0	Subregion 19 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
U RW SR20			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 20 in region 0 (write '1' to clear)																													
V RW SR21			NoAccess	0	Subregion 21 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
W RW SR22			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 22 in region 0 (write '1' to clear)																													
X RW SR23			NoAccess	0	Subregion 23 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
Y RW SR24			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 24 in region 0 (write '1' to clear)																													
Z RW SR25			NoAccess	0	Subregion 25 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
a RW SR26			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 26 in region 0 (write '1' to clear)																													
b RW SR27			NoAccess	0	Subregion 27 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
c RW SR28			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 28 in region 0 (write '1' to clear)																													
d RW SR29			NoAccess	0	Subregion 29 in region 0 (write '1' to clear)																													
			Access	1	No write access occurred in this subregion																													
e RW SR30			NoAccess	0	Write access(es) occurred in this subregion																													
			Access	1	Subregion 30 in region 0 (write '1' to clear)																													
f RW SR31			NoAccess	0	Subregion 31 in region 0 (write '1' to clear)																													

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id			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	
Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value	Id	RW	Field	Value			
A RW SR0																												Subregion 0 in region 0 (write '1' to clear)						
																															No read access occurred in this subregion			
																															Read access(es) occurred in this subregion			
B RW SR1																																Subregion 1 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
C RW SR2																																Subregion 2 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
D RW SR3																																Subregion 3 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
E RW SR4																																Subregion 4 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
F RW SR5																																Subregion 5 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
G RW SR6																																Subregion 6 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
H RW SR7																																Subregion 7 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
I RW SR8																																Subregion 8 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
J RW SR9																																Subregion 9 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		
K RW SR10																																Subregion 10 in region 0 (write '1' to clear)		
																																No read access occurred in this subregion		
																																Read access(es) occurred in this subregion		

## 45.1.9 PERREGION[1].SUBSTATWA

Address offset: 0x408

Source of event/interrupt in region 1, write access detected while corresponding subregion was enabled for watching



Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
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		
Id	RW	Field	Value Id																										Description					
U			NoAccess																									No write access occurred in this subregion						
			Access																									Write access(es) occurred in this subregion						
V			Subregion 20 in region 1 (write '1' to clear)																															
			NoAccess																									No write access occurred in this subregion						
W			Access																									Write access(es) occurred in this subregion						
			Subregion 22 in region 1 (write '1' to clear)																															
X			NoAccess																									Subregion 23 in region 1 (write '1' to clear)						
			Access																									Write access(es) occurred in this subregion						
Y			Subregion 24 in region 1 (write '1' to clear)																															
			NoAccess																								No write access occurred in this subregion							
Z			Access																								Write access(es) occurred in this subregion							
			Subregion 25 in region 1 (write '1' to clear)																															
a			NoAccess																								Subregion 26 in region 1 (write '1' to clear)							
			Access																								Write access(es) occurred in this subregion							
b			Subregion 27 in region 1 (write '1' to clear)																															
			NoAccess																								No write access occurred in this subregion							
c			Access																								Write access(es) occurred in this subregion							
			Subregion 28 in region 1 (write '1' to clear)																															
d			NoAccess																								No write access occurred in this subregion							
			Access																								Write access(es) occurred in this subregion							
e			Subregion 30 in region 1 (write '1' to clear)																															
			NoAccess																								No write access occurred in this subregion							
f			Access																								Write access(es) occurred in this subregion							
			Subregion 31 in region 1 (write '1' to clear)																															

#### **45.1.10 PERREGION[1].SUBSTATRA**

Address offset: 0x40C

Source of event/interrupt in region 1, read access detected while corresponding subregion was enabled for watching

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
A	RW	SR0	NoAccess	0		Subregion 0 in region 1 (write '1' to clear)																										
			Access	1		No read access occurred in this subregion																										
			NoAccess	0		Read access(es) occurred in this subregion																										
B	RW	SR1	NoAccess	0		Subregion 1 in region 1 (write '1' to clear)																										
			Access	1		No read access occurred in this subregion																										
			NoAccess	0		Read access(es) occurred in this subregion																										



Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
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		
Id	RW	Field	Value Id																										Description					
V			NoAccess																									No read access occurred in this subregion						
			Access																									Read access(es) occurred in this subregion						
W			Subregion 21 in region 1 (write '1' to clear)																															
			NoAccess																										No read access occurred in this subregion					
X			Access																									Read access(es) occurred in this subregion						
			Subregion 23 in region 1 (write '1' to clear)																															
Y			NoAccess																									No read access occurred in this subregion						
			Access																									Read access(es) occurred in this subregion						
Z			Subregion 25 in region 1 (write '1' to clear)																															
			NoAccess																									No read access occurred in this subregion						
a			Access																									Read access(es) occurred in this subregion						
			Subregion 26 in region 1 (write '1' to clear)																															
b			NoAccess																									No read access occurred in this subregion						
			Access																									Read access(es) occurred in this subregion						
c			Subregion 28 in region 1 (write '1' to clear)																															
			NoAccess																									No read access occurred in this subregion						
d			Access																									Read access(es) occurred in this subregion						
			Subregion 29 in region 1 (write '1' to clear)																															
e			NoAccess																									No read access occurred in this subregion						
			Access																									Read access(es) occurred in this subregion						
f			Subregion 31 in region 1 (write '1' to clear)																															
			NoAccess																									No read access occurred in this subregion						
			Access																									Read access(es) occurred in this subregion						

## 45.1.11 REGIONEN

Address offset: 0x510

Enable/disable regions watch

## 45.1.12 REGIONENSET

Address offset: 0x514

## Enable regions watch

### **45.1.13 REGIONENCLR**

Address offset: 0x518

## Disable regions watch

## 45.1.14 REGION[0].START

Address offset: 0x600

### Start address for region 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	START			Start address for region																											

## 45.1.15 REGION[0].END

Address offset: 0x604

End address of region 0

## 45.1.16 REGION[1].START

Address offset: 0x610

Start address for region 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	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	START			Start address for region																											

## 45.1.17 REGION[1].END

Address offset: 0x614

End address of region 1

## 45.1.18 REGION[2].START

Address offset: 0x620

## Start address for region 2

## 45.1.19 REGION[2].END

Address offset: 0x624

End address of region 2

## 45.1.20 REGION[3].START

Address offset: 0x630

### Start address for region 3

## 45.1.21 REGION[3].END

Address offset: 0x634

End address of region 3

## 45.1.22 PREGION[0].START

Address offset: 0x6C0

Reserved for future use

## 45.1.23 PREGION[0].END

Address offset: 0x6C4

Reserved for future use

## 45.1.24 PREGION[0].SUBS

Address offset: 0x6C8

## Subregions of region 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	
Id			RW	Field	Value Id																											Description			
A			RW	SR0																										Include or exclude subregion 0 in region					
					Exclude																									Exclude					
					Include																									Include					
B			RW	SR1																										Include or exclude subregion 1 in region					
					Exclude																									Exclude					
					Include																									Include					
C			RW	SR2																										Include or exclude subregion 2 in region					
					Exclude																									Exclude					
					Include																									Include					
D			RW	SR3																										Include or exclude subregion 3 in region					
					Exclude																									Exclude					
					Include																									Include					
E			RW	SR4																										Include or exclude subregion 4 in region					
					Exclude																									Exclude					
					Include																									Include					
F			RW	SR5																										Include or exclude subregion 5 in region					
					Exclude																									Exclude					
					Include																									Include					
G			RW	SR6																										Include or exclude subregion 6 in region					
					Exclude																									Exclude					
					Include																									Include					



## 45.1.25 PREGION[1].START

Address offset: 0x6D0

Reserved for future use

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	START			Reserved for future use																											

## 45.1.26 PREGION[1].END

Address offset: 0x6D4

Reserved for future use

## 45.1.27 PREGION[1].SUBS

Address offset: 0x6D8

## Subregions of region 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	
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		
Id	RW	Field	Value Id	Value																										Description				
B			Include	1																										Include				
C			Exclude	0																										Exclude				
D			Include	1																										Include				
E			Include or exclude subregion 1 in region																											Include or exclude subregion 2 in region				
F			Exclude	0																										Exclude				
G			Include	1																										Include				
H			Include or exclude subregion 3 in region																										Include or exclude subregion 4 in region					
I			Exclude	0																									Exclude					
J			Include	1																									Include					
K			Include or exclude subregion 5 in region																										Include or exclude subregion 6 in region					
L			Exclude	0																									Exclude					
M			Include	1																									Include					
N			Include or exclude subregion 7 in region																										Include or exclude subregion 8 in region					
O			Exclude	0																									Exclude					
P			Include	1																									Include					
Q			Include or exclude subregion 9 in region																										Include or exclude subregion 10 in region					
R			Exclude	0																									Exclude					
S			Include	1																									Include					
T			Include or exclude subregion 11 in region																										Include or exclude subregion 12 in region					
U			Exclude	0																									Exclude					
V			Include	1																									Include					
W			Include or exclude subregion 13 in region																										Include or exclude subregion 14 in region					
X			Exclude	0																									Exclude					
Y			Include	1																									Include					
Z			Include or exclude subregion 15 in region																										Include or exclude subregion 16 in region					
AA			Exclude	0																									Exclude					
AB			Include	1																									Include					
AC			Include or exclude subregion 17 in region																										Include or exclude subregion 18 in region					
AD			Exclude	0																									Exclude					
AE			Include	1																									Include					

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
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		
Id	RW	Field	Value Id																										Description					
T	RW	SR19																										Include or exclude subregion 19 in region						
			Exclude																										Exclude					
			Include																										Include					
U	RW	SR20																											Include or exclude subregion 20 in region					
			Exclude																										Exclude					
			Include																										Include					
V	RW	SR21																											Include or exclude subregion 21 in region					
			Exclude																										Exclude					
			Include																										Include					
W	RW	SR22																											Include or exclude subregion 22 in region					
			Exclude																										Exclude					
			Include																										Include					
X	RW	SR23																											Include or exclude subregion 23 in region					
			Exclude																										Exclude					
			Include																										Include					
Y	RW	SR24																											Include or exclude subregion 24 in region					
			Exclude																										Exclude					
			Include																										Include					
Z	RW	SR25																											Include or exclude subregion 25 in region					
			Exclude																										Exclude					
			Include																										Include					
a	RW	SR26																											Include or exclude subregion 26 in region					
			Exclude																										Exclude					
			Include																										Include					
b	RW	SR27																											Include or exclude subregion 27 in region					
			Exclude																										Exclude					
			Include																										Include					
c	RW	SR28																											Include or exclude subregion 28 in region					
			Exclude																										Exclude					
			Include																										Include					
d	RW	SR29																											Include or exclude subregion 29 in region					
			Exclude																										Exclude					
			Include																										Include					
e	RW	SR30																											Include or exclude subregion 30 in region					
			Exclude																										Exclude					
			Include																										Include					
f	RW	SR31																											Include or exclude subregion 31 in region					
			Exclude																										Exclude					
			Include																										Include					

# 46 EGU — Event generator unit

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

Listed here are the main EGU features:

- Enables SW triggering of interrupts
- 6 EGU instances – separate interrupt vectors
- Up to 16 separate event flags per interrupt for multiplexing

The EGU implements a set of tasks which can individually be triggered to generate the corresponding event, i.e., the corresponding event for TASKS\_TRIGGER[n] is EVENTS\_TRIGGERED[n].

**Table 113: EGU configuration**

EGU instance	Number of event flags
0-5	16

## 46.1 Registers

**Table 114: Instances**

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

**Table 115: Register Overview**

Register	Offset	Description
TASKS_TRIGGER[0]	0x000	Trigger 0 for triggering the corresponding TRIGGERED[0] event
TASKS_TRIGGER[1]	0x004	Trigger 1 for triggering the corresponding TRIGGERED[1] event
TASKS_TRIGGER[2]	0x008	Trigger 2 for triggering the corresponding TRIGGERED[2] event
TASKS_TRIGGER[3]	0x00C	Trigger 3 for triggering the corresponding TRIGGERED[3] event
TASKS_TRIGGER[4]	0x010	Trigger 4 for triggering the corresponding TRIGGERED[4] event
TASKS_TRIGGER[5]	0x014	Trigger 5 for triggering the corresponding TRIGGERED[5] event
TASKS_TRIGGER[6]	0x018	Trigger 6 for triggering the corresponding TRIGGERED[6] event
TASKS_TRIGGER[7]	0x01C	Trigger 7 for triggering the corresponding TRIGGERED[7] event
TASKS_TRIGGER[8]	0x020	Trigger 8 for triggering the corresponding TRIGGERED[8] event
TASKS_TRIGGER[9]	0x024	Trigger 9 for triggering the corresponding TRIGGERED[9] event
TASKS_TRIGGER[10]	0x028	Trigger 10 for triggering the corresponding TRIGGERED[10] event
TASKS_TRIGGER[11]	0x02C	Trigger 11 for triggering the corresponding TRIGGERED[11] event
TASKS_TRIGGER[12]	0x030	Trigger 12 for triggering the corresponding TRIGGERED[12] event
TASKS_TRIGGER[13]	0x034	Trigger 13 for triggering the corresponding TRIGGERED[13] event
TASKS_TRIGGER[14]	0x038	Trigger 14 for triggering the corresponding TRIGGERED[14] event
TASKS_TRIGGER[15]	0x03C	Trigger 15 for triggering the corresponding TRIGGERED[15] event
EVENTS_TRIGGERED[0]	0x100	Event number 0 generated by triggering the corresponding TRIGGER[0] task
EVENTS_TRIGGERED[1]	0x104	Event number 1 generated by triggering the corresponding TRIGGER[1] task
EVENTS_TRIGGERED[2]	0x108	Event number 2 generated by triggering the corresponding TRIGGER[2] task
EVENTS_TRIGGERED[3]	0x10C	Event number 3 generated by triggering the corresponding TRIGGER[3] task
EVENTS_TRIGGERED[4]	0x110	Event number 4 generated by triggering the corresponding TRIGGER[4] task

Register	Offset	Description
EVENTS_TRIGGERED[5]	0x114	Event number 5 generated by triggering the corresponding TRIGGER[5] task
EVENTS_TRIGGERED[6]	0x118	Event number 6 generated by triggering the corresponding TRIGGER[6] task
EVENTS_TRIGGERED[7]	0x11C	Event number 7 generated by triggering the corresponding TRIGGER[7] task
EVENTS_TRIGGERED[8]	0x120	Event number 8 generated by triggering the corresponding TRIGGER[8] task
EVENTS_TRIGGERED[9]	0x124	Event number 9 generated by triggering the corresponding TRIGGER[9] task
EVENTS_TRIGGERED[10]	0x128	Event number 10 generated by triggering the corresponding TRIGGER[10] task
EVENTS_TRIGGERED[11]	0x12C	Event number 11 generated by triggering the corresponding TRIGGER[11] task
EVENTS_TRIGGERED[12]	0x130	Event number 12 generated by triggering the corresponding TRIGGER[12] task
EVENTS_TRIGGERED[13]	0x134	Event number 13 generated by triggering the corresponding TRIGGER[13] task
EVENTS_TRIGGERED[14]	0x138	Event number 14 generated by triggering the corresponding TRIGGER[14] task
EVENTS_TRIGGERED[15]	0x13C	Event number 15 generated by triggering the corresponding TRIGGER[15] task
<b>INTEN</b>	0x300	Enable or disable interrupt
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt

## 46.1.1 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id		P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A																
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	TRIGGERED0				Enable or disable interrupt for TRIGGERED[0] event																											
						See <a href="#">EVENTS_TRIGGERED[0]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
B	RW	TRIGGERED1				Enable or disable interrupt for TRIGGERED[1] event																											
						See <a href="#">EVENTS_TRIGGERED[1]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
C	RW	TRIGGERED2				Enable or disable interrupt for TRIGGERED[2] event																											
						See <a href="#">EVENTS_TRIGGERED[2]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
D	RW	TRIGGERED3				Enable or disable interrupt for TRIGGERED[3] event																											
						See <a href="#">EVENTS_TRIGGERED[3]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
E	RW	TRIGGERED4				Enable or disable interrupt for TRIGGERED[4] event																											
						See <a href="#">EVENTS_TRIGGERED[4]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
F	RW	TRIGGERED5				Enable or disable interrupt for TRIGGERED[5] event																											
						See <a href="#">EVENTS_TRIGGERED[5]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
G	RW	TRIGGERED6				Enable or disable interrupt for TRIGGERED[6] event																											
						See <a href="#">EVENTS_TRIGGERED[6]</a>																											
		Disabled	0			Disable																											
		Enabled	1			Enable																											
H	RW	TRIGGERED7				Enable or disable interrupt for TRIGGERED[7] event																											
						See <a href="#">EVENTS_TRIGGERED[7]</a>																											
		Disabled	0			Disable																											

## **46.1.2 INTENSET**

Address offset: 0x304

## Enable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
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	
<b>Id</b>			<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Description</b>																											
					Disabled	Read: Disabled																											
					Enabled	Read: Enabled																											
C	RW	TRIGGERED2				Write '1' to Enable interrupt for TRIGGERED[2] event																											
						See <a href="#">EVENTS_TRIGGERED[2]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
D	RW	TRIGGERED3				Write '1' to Enable interrupt for TRIGGERED[3] event																											
						See <a href="#">EVENTS_TRIGGERED[3]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
E	RW	TRIGGERED4				Write '1' to Enable interrupt for TRIGGERED[4] event																											
						See <a href="#">EVENTS_TRIGGERED[4]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
F	RW	TRIGGERED5				Write '1' to Enable interrupt for TRIGGERED[5] event																											
						See <a href="#">EVENTS_TRIGGERED[5]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
G	RW	TRIGGERED6				Write '1' to Enable interrupt for TRIGGERED[6] event																											
						See <a href="#">EVENTS_TRIGGERED[6]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
H	RW	TRIGGERED7				Write '1' to Enable interrupt for TRIGGERED[7] event																											
						See <a href="#">EVENTS_TRIGGERED[7]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
I	RW	TRIGGERED8				Write '1' to Enable interrupt for TRIGGERED[8] event																											
						See <a href="#">EVENTS_TRIGGERED[8]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
J	RW	TRIGGERED9				Write '1' to Enable interrupt for TRIGGERED[9] event																											
						See <a href="#">EVENTS_TRIGGERED[9]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
K	RW	TRIGGERED10				Write '1' to Enable interrupt for TRIGGERED[10] event																											
						See <a href="#">EVENTS_TRIGGERED[10]</a>																											
					Set	Enable																											
					Disabled	Read: Disabled																											
L	RW	TRIGGERED11				Write '1' to Enable interrupt for TRIGGERED[11] event																											
						See <a href="#">EVENTS_TRIGGERED[11]&lt;/a</a>																											

Bit number		31 30 29 28 27 26 25 24 23 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			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
M	RW	TRIGGERED12	Enabled	1	Read: Enabled Write '1' to Enable interrupt for TRIGGERED[12] event  See <a href="#">EVENTS_TRIGGERED[12]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	TRIGGERED13			Write '1' to Enable interrupt for TRIGGERED[13] event  See <a href="#">EVENTS_TRIGGERED[13]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	TRIGGERED14			Write '1' to Enable interrupt for TRIGGERED[14] event  See <a href="#">EVENTS_TRIGGERED[14]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	TRIGGERED15			Write '1' to Enable interrupt for TRIGGERED[15] event  See <a href="#">EVENTS_TRIGGERED[15]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

### 46.1.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
Id		P O N M L K J I H G F E D C B A			
<b>Reset 0x00000000</b>		0 0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
A	RW	TRIGGERED0			Write '1' to Disable interrupt for TRIGGERED[0] event  See <a href="#">EVENTS_TRIGGERED[0]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	TRIGGERED1			Write '1' to Disable interrupt for TRIGGERED[1] event  See <a href="#">EVENTS_TRIGGERED[1]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	TRIGGERED2			Write '1' to Disable interrupt for TRIGGERED[2] event  See <a href="#">EVENTS_TRIGGERED[2]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TRIGGERED3			Write '1' to Disable interrupt for TRIGGERED[3] event  See <a href="#">EVENTS_TRIGGERED[3]</a>
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TRIGGERED4			Write '1' to Disable interrupt for TRIGGERED[4] event

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	P O N M L K J I H G F E D C B A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_TRIGGERED[4]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
F	RW	TRIGGERED5			Write '1' to Disable interrupt for TRIGGERED[5] event
					See <a href="#">EVENTS_TRIGGERED[5]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
G	RW	TRIGGERED6			Write '1' to Disable interrupt for TRIGGERED[6] event
					See <a href="#">EVENTS_TRIGGERED[6]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
H	RW	TRIGGERED7			Write '1' to Disable interrupt for TRIGGERED[7] event
					See <a href="#">EVENTS_TRIGGERED[7]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
I	RW	TRIGGERED8			Write '1' to Disable interrupt for TRIGGERED[8] event
					See <a href="#">EVENTS_TRIGGERED[8]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
J	RW	TRIGGERED9			Write '1' to Disable interrupt for TRIGGERED[9] event
					See <a href="#">EVENTS_TRIGGERED[9]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
K	RW	TRIGGERED10			Write '1' to Disable interrupt for TRIGGERED[10] event
					See <a href="#">EVENTS_TRIGGERED[10]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
L	RW	TRIGGERED11			Write '1' to Disable interrupt for TRIGGERED[11] event
					See <a href="#">EVENTS_TRIGGERED[11]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
M	RW	TRIGGERED12			Write '1' to Disable interrupt for TRIGGERED[12] event
					See <a href="#">EVENTS_TRIGGERED[12]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
N	RW	TRIGGERED13			Write '1' to Disable interrupt for TRIGGERED[13] event
					See <a href="#">EVENTS_TRIGGERED[13]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
O	RW	TRIGGERED14			Write '1' to Disable interrupt for TRIGGERED[14] event
					See <a href="#">EVENTS_TRIGGERED[14]</a>

Bit number	31 30 29 28 27 26 25 24 23 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				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
P	RW	TRIGGERED15			Write '1' to Disable interrupt for TRIGGERED[15] event See <a href="#">EVENTS_TRIGGERED[15]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

## 46.2 Electrical specification

### 46.2.1 EGU Electrical Specification

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

# 47 PWM — Pulse width modulation

The PWM module enables the generation of pulse width modulated signals on GPIO. The module implements an up or up-and-down counter with four PWM channels that drive assigned GPIOs.

Three PWM modules can provide up to 12 PWM channels with individual frequency control in groups of up to four channels. Furthermore, a built-in decoder and EasyDMA capabilities make it possible to manipulate the PWM duty cycles without CPU intervention. Arbitrary duty-cycle sequences are read from Data RAM and can be chained to implement ping-pong buffering or repeated into complex loops.

Listed here are the main features of one PWM module:

- Fixed PWM base frequency with programmable clock divider
- Up to four PWM channels with individual polarity and duty-cycle values
- Edge or center-aligned pulses across PWM channels
- Multiple duty-cycle arrays (sequences) defined in Data RAM
- Autonomous and glitch-free update of duty cycle values directly from memory through EasyDMA
- Change of polarity, duty-cycle, and base frequency possibly on every PWM period
- Data RAM sequences can be repeated or connected into loops

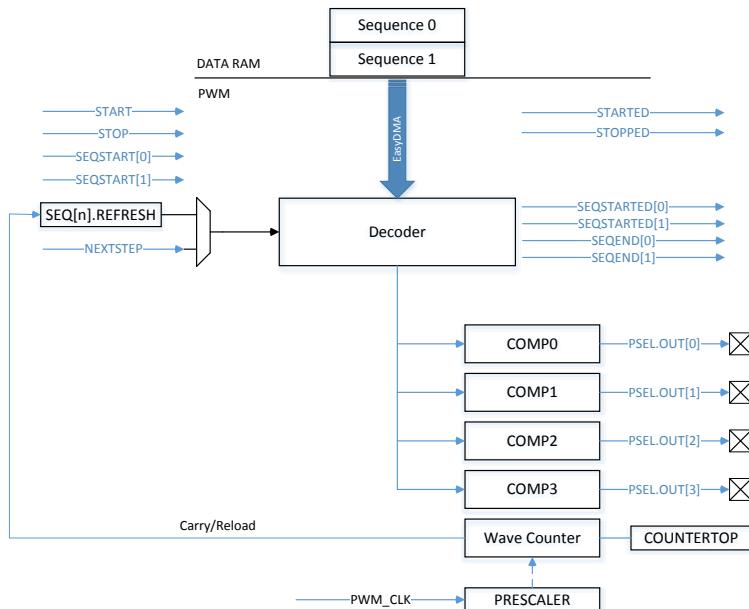


Figure 141: PWM Module

## 47.1 Wave counter

The wave counter is responsible for generating the pulses at a duty-cycle that depends on the compare values, and at a frequency that depends on COUNTERTOP.

There is one common 15-bit counter with four compare channels. Thus, all four channels will share the same period (PWM frequency), but can have individual duty-cycle and polarity. The polarity is set by the value read from RAM (see [Figure 144: Decoder memory access modes](#) on page 498), while the MODE register controls if the counter counts up, or up and down. The timer top value is controlled by the COUNTERTOP register. This register value in conjunction with the selected PRESCALER of the PWM\_CLK will result in a given PWM period. A COUNTERTOP value smaller than the compare setting will result in a state where no PWM edges are generated. Respectively, OUT[n] is held high, given that the polarity is set to FallingEdge. All the compare registers are internal and can only be configured through the decoder presented later.

COUNTERTOP can be safely written at any time. It will get sampled following a START task. If DECODER.LOAD is anything else than WaveForm, it will also get sampled following a STARTSEQ[n] task,

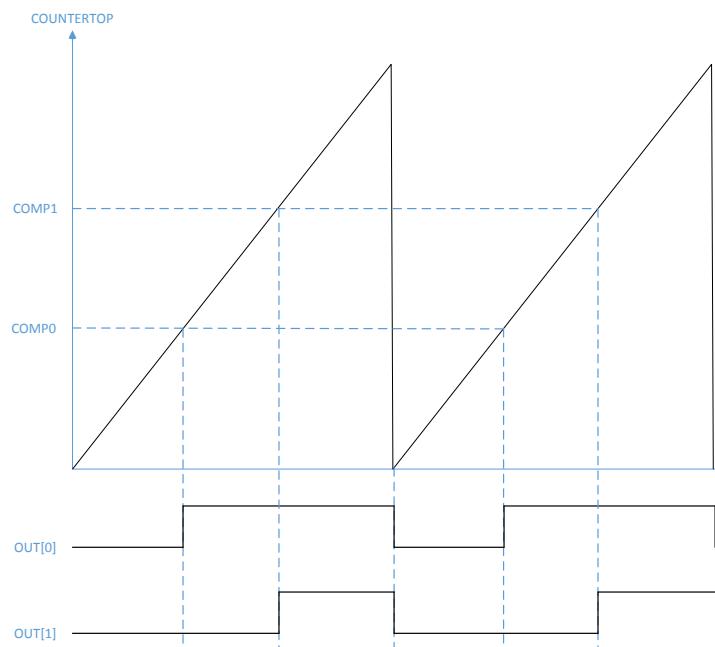
and when loading a new value from RAM during a sequence playback. If DECODER.LOAD=WaveForm, the register value is ignored, and taken from RAM instead (see [Decoder with EasyDMA](#) on page 498 below).

**Figure 142: PWM up counter example - FallingEdge polarity** on page 496 shows the counter operating in up (MODE=PWM\_MODE\_Up) mode with three PWM channels with the same frequency but different duty cycle. The counter is automatically reset to zero when COUNTERTOP is reached and OUT[n] will invert. OUT[n] is held low if the compare value is 0 and held high respectively if set to COUNTERTOP given that the polarity is set to FallingEdge. Running in up counter mode will result in pulse widths that are edge-aligned. See the code example below:

```

uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY,
    PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE = (PWM_ENABLE_ENABLED << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER = (PWM_PRESCALER_PRESCALER_DIV_1 <<
    PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
    (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
    PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```



**Figure 142: PWM up counter example - FallingEdge polarity**

In up counting mode, the following formula can be used to compute PWM period and step size:

PWM period:  $T_{\text{PWM}}(\text{Up}) = T_{\text{PWM\_CLK}} * \text{COUNTERTOP}$

Step width/Resolution:  $T_{\text{steps}} = T_{\text{PWM\_CLK}}$

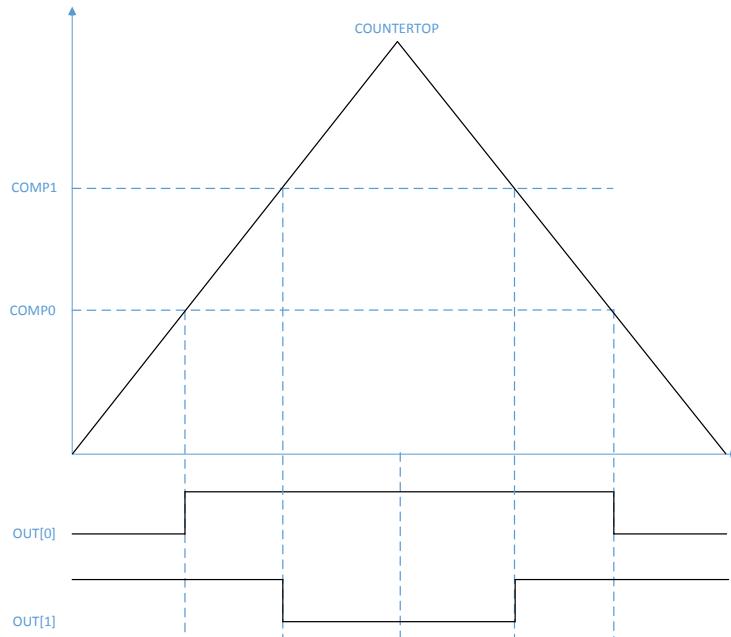
[Figure 143: PWM up-and-down counter example](#) on page 497 shows the counter operating in up and down mode with (MODE=PWM\_MODE\_UpAndDown) two PWM channels with the same frequency but different duty cycle and output polarity. The counter starts decrementing to zero when COUNTERTOP is reached and will invert the OUT[n] when compare value is hit for the second time. This results in a set of pulses that are center-aligned.

```

uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY,
  PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
  (PWM_PSEL_OUT_CONNECT_Connected <<
   PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
  (PWM_PSEL_OUT_CONNECT_Connected <<
   PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE = (PWM_MODE_UPDOWN_UpAndDown << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER = (PWM_PRESCALER_PRESCALER_DIV_1 <<
  PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
  (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
  PWM_SEQ_CNT_CNT_Pos);

NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```



**Figure 143: PWM up-and-down counter example**

In up-and-down counting modes, the following formula can be used to compute PWM period and step size:

$$T_{\text{PWM}}(\text{Up And Down}) = T_{\text{PWM\_CLK}} * 2 * \text{COUNTERTOP}$$

$$\text{Step width/Resolution: } T_{\text{steps}} = T_{\text{PWM\_CLK}} * 2$$

## 47.2 Decoder with EasyDMA

The decoder uses EasyDMA to take PWM parameters stored in Data RAM by ways of EasyDMA and updates the internal compare registers of the wave counter based on the mode of operation.

The mentioned PWM parameters are organized into a sequence containing at least one half word (16 bit). Its most significant bit[15] denotes the polarity of the OUT[n] while bit[14:0] is the 15-bit compare value. See below for further details of these RAM defined registers.

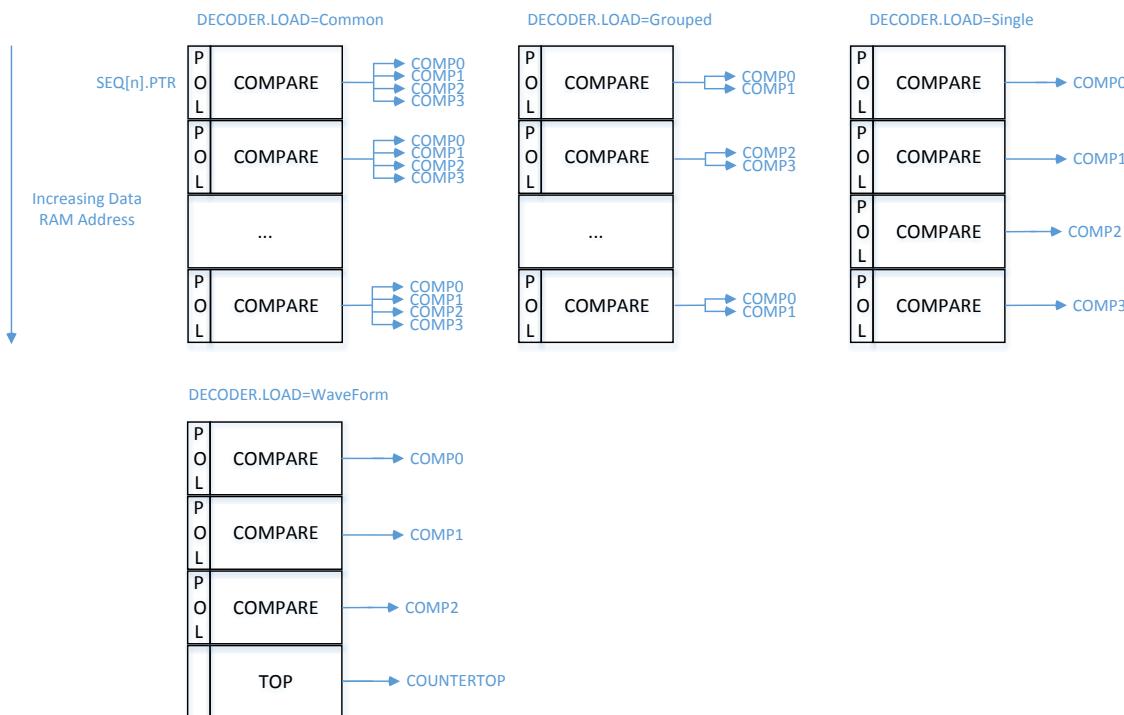
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																		B	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>Value</b>	<b>Description</b>																											
A	RW	COMPARE				Duty cycle setting - value loaded to internal compare register																											
B	RW	POLARITY				Edge polarity of GPIO.																											
		RisingEdge	0			First edge within the PWM period is rising																											
		FallingEdge	1			First edge within the PWM period is falling																											

The DECODER register controls how the RAM content is interpreted and loaded to the internal compare registers. The LOAD field can be used to control if the RAM values are loaded to all compare channels - or alternatively to update a group or all channels with individual values. [Figure 144: Decoder memory access modes](#) on page 498 illustrates how the parameters stored in RAM are organized and routed to the various compare channels in the different modes.

A special mode of operation is available when DECODER.LOAD is set to WaveForm. In this mode, up to three PWM channels can be enabled - OUT[0] to OUT[2]. In RAM, four values are loaded at a time: the first, second and third location are used to load the values, and the fourth RAM location is used to load the COUNTERTOP register. This way one can have up to three PWM channels with a frequency base that changes on a per PWM period basis. This mode of operation is useful for arbitrary wave form generation in applications such as LED lighting.

The register SEQ[n].REFRESH=N (one per sequence n=0 or 1) will instruct a new RAM stored pulse width value on every (N+1)<sup>th</sup> PWM period. Setting the register to zero will result in a new duty cycle update every PWM period as long as the minimum PWM period is observed.

Note that registers SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored when DECODER.MODE=NextStep . The next value is loaded upon receiving every NEXTSTEP task.



**Figure 144: Decoder memory access modes**

SEQ[n].PTR is the pointer used to fetch COMPARE values from RAM. If the SEQ[n].PTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 23 for more information about the different memory regions.

After the SEQ[n].PTR is set to the desired RAM location, the SEQ[n].CNT register must be set to the number of 16-bit half words in the sequence. It is important to observe that the Grouped and Single modes require one half word per group or one half word per channel respectively, and thus increases RAM size occupation. If PWM generation was not running yet at that point, sending the SEQSTART[n] task will load the first value from RAM, then start the PWM generation. A SEQSTARTED[n] event is generated as soon as the EasyDMA has read the first PWM parameter from RAM and the wave counter has started executing it. When LOOP.CNT=0, sequence n=0 or 1 is played back once. After the last value in the sequence has been loaded and started executing, a SEQEND[n] event is generated. The PWM generation will then continue with the last loaded value. See [Figure 145: Simple sequence example](#) on page 500 for an example of such simple playback.

To completely stop the PWM generation and force the associated pins to a defined state, a STOP task can be fired at any time. A STOPPED event is generated when the PWM generation has stopped at the end of currently running PWM period, and the pins go into their idle state as defined in GPIO->OUT. PWM generation can then only be restarted through a SEQSTART[n] task. SEQSTART[n] will resume PWM generation after having loaded the first value from the RAM buffer defined in the SEQ[n].PTR register.

The table below provides indication of when specific registers get sampled by the hardware. Care should be taken when updating these registers to avoid values to be applied earlier than expected.

**Table 116: When to safely update PWM registers**

Register	Taken into account by hardware	Recommended (safe) update
SEQ[n].PTR	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[n].CNT	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[0].ENDDELAY	When sending the SEQSTART[0] task  Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [0] through a SEQSTART[0] task  When no more value from sequence [0] gets loaded from RAM (indicated by the SEQEND[0] event)
SEQ[1].ENDDELAY	When sending the SEQSTART[1] task  Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [1] through a SEQSTART[1] task  When no more value from sequence [1] gets loaded from RAM (indicated by the SEQEND[1] event)  At any time during sequence [0] (which starts when the SEQSTARTED[1] event is fired)
SEQ[0].REFRESH	When sending the SEQSTART[0] task  Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [0] through a SEQSTART[0] task  At any time during sequence [1] (which starts when the SEQSTARTED[1] event is fired)
SEQ[1].REFRESH	When sending the SEQSTART[1] task  Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [1] through a SEQSTART[1] task  At any time during sequence [0] (which starts when the SEQSTARTED[0] event is fired)
COUNTERTOP	In DECODER.LOAD=WaveForm: this register is ignored.  In all other LOAD modes: at the end of current PWM period (indicated by the PWMPERIODEND event)	Before starting PWM generation through a SEQSTART[n] task  After a STOP task has been issued, and the STOPPED event has been received.
MODE	Immediately	Before starting PWM generation through a SEQSTART[n] task  After a STOP task has been issued, and the STOPPED event has been received.
DECODER	Immediately	Before starting PWM generation through a SEQSTART[n] task  After a STOP task has been issued, and the STOPPED event has been received.
PRESCALER	Immediately	Before starting PWM generation through a SEQSTART[n] task  After a STOP task has been issued, and the STOPPED event has been received.
LOOP	Immediately	Before starting PWM generation through a SEQSTART[n] task  After a STOP task has been issued, and the STOPPED event has been received.
PSEL.OUT[n]	Immediately	Before enabling the PWM instance through the ENABLE register

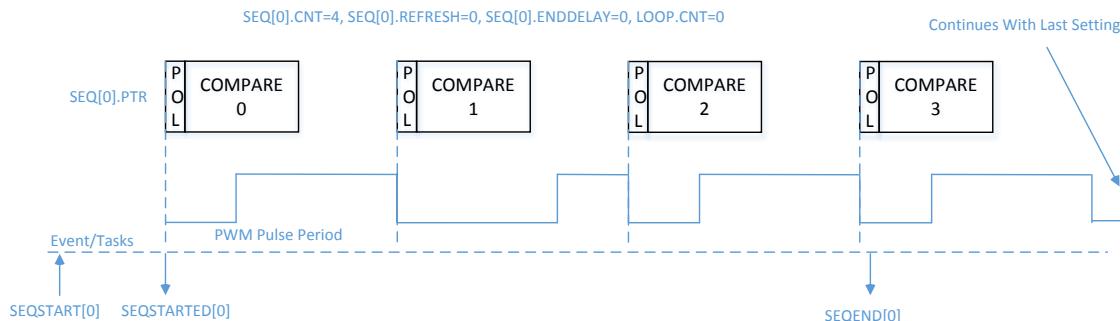
**Important:** SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored at the end of a complex sequence, indicated by a LOOPSDONE event. The reason for this is that the last value loaded from RAM is maintained until further action from software (restarting a new sequence, or stopping PWM generation).

[Figure 145: Simple sequence example](#) on page 500 depicts the source code used for configuration and timing details in a sequence where only sequence 0 is used and only run once with a new PWM duty cycle for each period.

```

NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
     PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
    PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER     = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
    (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR   = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT   = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
    PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```



**Figure 145: Simple sequence example**

A more complex example is shown in [Figure 146: Example using two sequences](#) on page 501, where LOOP.CNT>0 . In this case, an automated playback takes place, consisting of SEQ[0], delay 0, SEQ[1], delay 1, then again SEQ[0], etc. The user can choose to start a complex playback with SEQ[0] or SEQ[1] through sending the SEQSTART[0] or SEQSTART[1] task.

The complex playback always ends with delay 1.

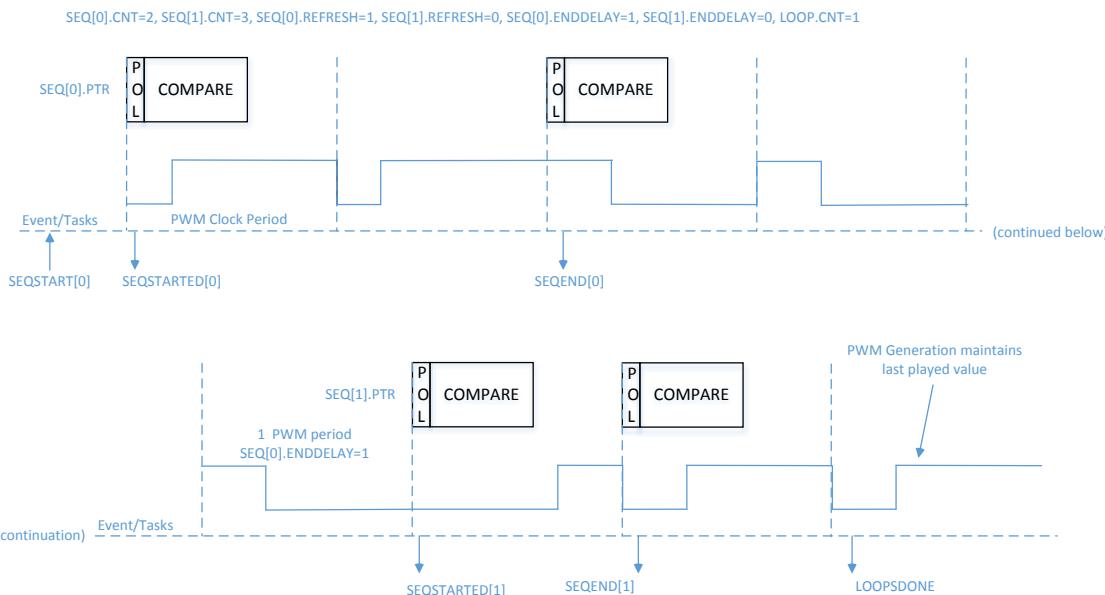
The two sequences 0 and 1 are defined with address of values tables in Data RAM (pointed by SEQ[n].PTR) and respective buffer size (SEQ[n].CNT). The rate at which a new value is loaded is defined individually for each sequence by SEQ[n].REFRESH . The chaining of sequence 1 following sequence 0 is implicit, the LOOP.CNT register allows the chaining of sequence 1 to sequence 0 for a determined number of times. In other words, it allows to repeat a complex sequence a number of times in a fully automated way.

In the example below, sequence 0 is defined with SEQ[0].REFRESH set to one - that means that a new PWM duty cycle is pushed every second PWM period. This complex sequence is started with the SEQSTART[0] task, so SEQ[0] is played first. Since SEQ[0].ENDDELAY=1 there will be one PWM period delay between last period on sequence 0 and the first period on sequence 1. Since SEQ[1].ENDDELAY=0 there is no delay 1, so SEQ[0] would be started immediately after the end of SEQ[1]. However, as LOOP.CNT is one, the playback stops after having played only once SEQ[1], and both SEQEND[1] and LOOPSDONE are generated (their order is not guaranteed in this case).

```

NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                           (PWM_PSEL_OUT_CONNECT_Connected <<
                            PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                           PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP         = (1 << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER     = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                           (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR   = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT   = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                           PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 1;
NRF_PWM0->SEQ[0].ENDDELAY = 1;
NRF_PWM0->SEQ[1].PTR   = ((uint32_t)(seq1_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[1].CNT   = ((sizeof(seq1_ram) / sizeof(uint16_t)) <<
                           PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[1].REFRESH = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```



**Figure 146: Example using two sequences**

The decoder can also be configured to asynchronously load a new PWM duty cycle. If the DECODER.MODE register is set to NextStep - then the NEXTSTEP task will cause an update of the internal compare registers on the next PWM period.

The figures below provide an overview of each part of an arbitrary sequence, in various modes (LOOP.CNT=0 and LOOP.CNT>0). In particular are represented:

- Initial and final duty cycle on the PWM output(s)
- Chaining of SEQ[0] and SEQ[1] if LOOP.CNT>0
- Influence of registers on the sequence
- Events fired during a sequence
- DMA activity (loading of next value and applying it to the output(s))

Note that the single-shot example applies also to SEQ[1], only SEQ[0] is represented for simplicity.

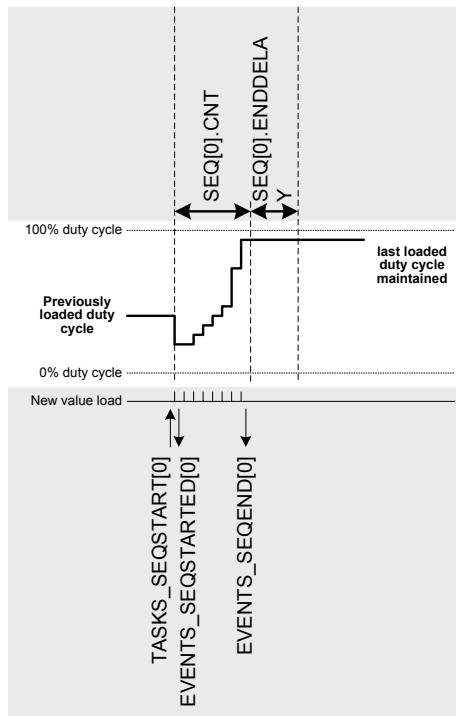


Figure 147: Single shot (`LOOP.CNT=0`)

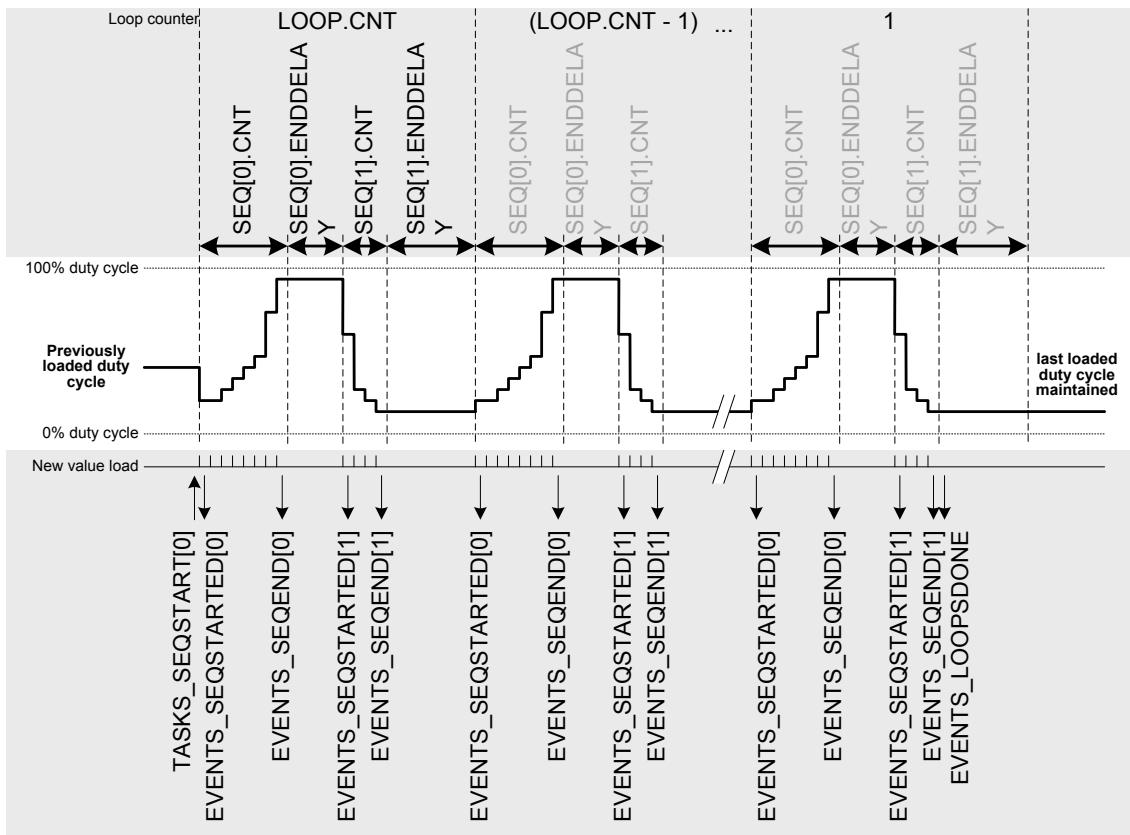
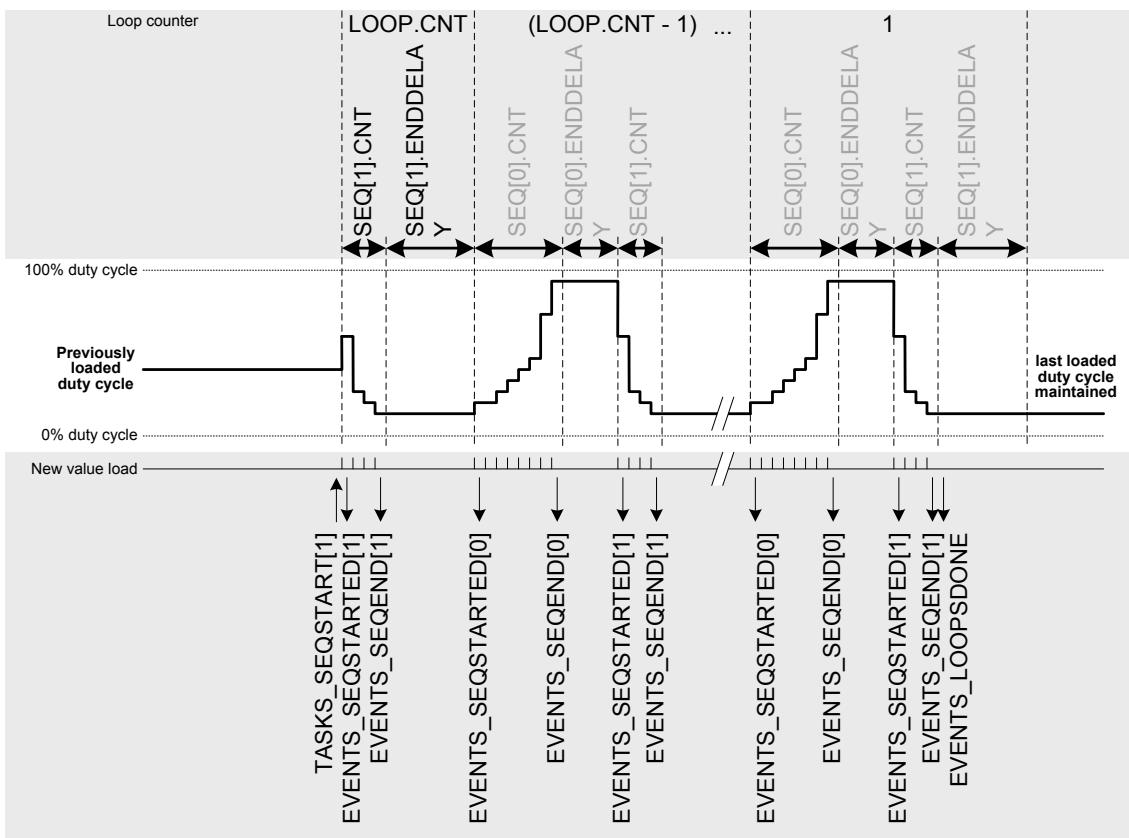


Figure 148: Complex sequence (`LOOP.CNT>0`) starting with `SEQ[0]`



**Figure 149: Complex sequence (LOOP.CNT>0) starting with SEQ[1]**

Note that if a sequence is in use in a simple or complex sequence, it must have a length of  $\text{SEQ}[n].\text{CNT} > 0$ .

### 47.3 Limitations

The previous compare value will be repeated if the PWM period is selected to be shorter than the time it takes for the EasyDMA to fetch from RAM and update the internal compare registers.

This is to ensure a glitch-free operation even if very short PWM periods are chosen.

### 47.4 Pin configuration

The OUT[n] ( $n=0..3$ ) signals associated to each channel of the PWM module are mapped to physical pins according to the configuration specified in the respective PSEL.OUT[n] registers. If a PSEL.OUT[n].CONNECT is set to Disconnected, the associated PWM module signal will not be connected to any physical pins.

The PSEL.OUT[n] registers and their configurations are only used as long as the PWM module is enabled and PWM generation is active (wave counter started), and retained only as long as the device is in System ON mode, see [POWER](#) chapter for more information about power modes.

To ensure correct behaviour in the PWM module, the pins used by the PWM module must be configured in the GPIO peripheral as described in [Table 117: Recommended GPIO configuration before starting PWM generation](#) on page 504 before enabling the PWM module. The pins' idle state is defined by the OUT registers in the GPIO module. This is to ensure that the pins used by the PWM module are driven correctly, if PWM generation is stopped through a STOP task, the PWM module itself is temporarily disabled, or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected IOs as long as the PWM module is supposed to be connected to an external PWM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behaviour.

**Table 117: Recommended GPIO configuration before starting PWM generation**

PWM signal	PWM pin	Direction	Output value	Comment
OUT[n]	As specified in PSEL.OUT[n] (n=0..3)	Output	0	Idle state defined in GPIO->OUT

## 47.5 Registers

**Table 118: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x4001C000	PWM	PWM0	Pulse Width Modulation Unit 0	
0x40021000	PWM	PWM1	Pulse Width Modulation Unit 1	
0x40022000	PWM	PWM2	Pulse Width Modulation Unit 2	

**Table 119: Register Overview**

Register	Offset	Description
TASKS_STOP	0x004	Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback
TASKS_SEQSTART[0]	0x008	Loads the first PWM value on all enabled channels from sequence 0, and starts playing that sequence at the rate defined in SEQ[0]REFRESH and/or DECODER.MODE. Causes PWM generation to start it was not running.
TASKS_SEQSTART[1]	0x00C	Loads the first PWM value on all enabled channels from sequence 1, and starts playing that sequence at the rate defined in SEQ[1]REFRESH and/or DECODER.MODE. Causes PWM generation to start it was not running.
TASKS_NEXSTEP	0x010	Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start it was not running.
EVENTS_STOPPED	0x104	Response to STOP task, emitted when PWM pulses are no longer generated
EVENTS_SEQSTARTED[0]	0x108	First PWM period started on sequence 0
EVENTS_SEQSTARTED[1]	0x10C	First PWM period started on sequence 1
EVENTS_SEQEND[0]	0x110	Emitted at end of every sequence 0, when last value from RAM has been applied to wave counter
EVENTS_SEQEND[1]	0x114	Emitted at end of every sequence 1, when last value from RAM has been applied to wave counter
EVENTS_PWMPERIODEN	0x118	Emitted at the end of each PWM period
EVENTS_LOOPSDONE	0x11C	Concatenated sequences have been played the amount of times defined in LOOP.CNT
SHORTS	0x200	Shortcut register
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	PWM module enable register
MODE	0x504	Selects operating mode of the wave counter
COUNTERTOP	0x508	Value up to which the pulse generator counter counts
PRESCALER	0x50C	Configuration for PWM_CLK
DECODER	0x510	Configuration of the decoder
LOOP	0x514	Amount of playback of a loop
SEQ[0].PTR	0x520	Beginning address in Data RAM of this sequence
SEQ[0].CNT	0x524	Amount of values (duty cycles) in this sequence
SEQ[0].REFRESH	0x528	Amount of additional PWM periods between samples loaded into compare register
SEQ[0].ENDDELAY	0x52C	Time added after the sequence
SEQ[1].PTR	0x540	Beginning address in Data RAM of this sequence
SEQ[1].CNT	0x544	Amount of values (duty cycles) in this sequence
SEQ[1].REFRESH	0x548	Amount of additional PWM periods between samples loaded into compare register
SEQ[1].ENDDELAY	0x54C	Time added after the sequence
PSEL.OUT[0]	0x560	Output pin select for PWM channel 0
PSEL.OUT[1]	0x564	Output pin select for PWM channel 1
PSEL.OUT[2]	0x568	Output pin select for PWM channel 2
PSEL.OUT[3]	0x56C	Output pin select for PWM channel 3

## 47.5.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0						
Id		E D C B A						
<b>Reset 0x00000000</b>		0 0						
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Description</b>
A	RW	SEQEND0_STOP						Shortcut between SEQEND[0] event and STOP task
								See <a href="#">EVENTS_SEQEND[0]</a> and <a href="#">TASKS_STOP</a>
			Disabled				0	Disable shortcut
			Enabled				1	Enable shortcut
B	RW	SEQEND1_STOP						Shortcut between SEQEND[1] event and STOP task
								See <a href="#">EVENTS_SEQEND[1]</a> and <a href="#">TASKS_STOP</a>
			Disabled				0	Disable shortcut
			Enabled				1	Enable shortcut
C	RW	LOOPSDONE_SEQSTART0						Shortcut between LOOPSDONE event and SEQSTART[0] task
								See <a href="#">EVENTS_LOOPSDONE</a> and <a href="#">TASKS_SEQSTART[0]</a>
			Disabled				0	Disable shortcut
			Enabled				1	Enable shortcut
D	RW	LOOPSDONE_SEQSTART1						Shortcut between LOOPSDONE event and SEQSTART[1] task
								See <a href="#">EVENTS_LOOPSDONE</a> and <a href="#">TASKS_SEQSTART[1]</a>
			Disabled				0	Disable shortcut
			Enabled				1	Enable shortcut
E	RW	LOOPSDONE_STOP						Shortcut between LOOPSDONE event and STOP task
								See <a href="#">EVENTS_LOOPSDONE</a> and <a href="#">TASKS_STOP</a>
			Disabled				0	Disable shortcut
			Enabled				1	Enable shortcut

## 47.5.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0						
Id		H G F E D C B						
<b>Reset 0x00000000</b>		0 0						
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value</b>	<b>Description</b>
B	RW	STOPPED						Enable or disable interrupt for STOPPED event
								See <a href="#">EVENTS_STOPPED</a>
			Disabled				0	Disable
			Enabled				1	Enable
C	RW	SEQSTARTED0						Enable or disable interrupt for SEQSTARTED[0] event
								See <a href="#">EVENTS_SEQSTARTED[0]</a>
			Disabled				0	Disable
			Enabled				1	Enable
D	RW	SEQSTARTED1						Enable or disable interrupt for SEQSTARTED[1] event
								See <a href="#">EVENTS_SEQSTARTED[1]</a>
			Disabled				0	Disable
			Enabled				1	Enable
E	RW	SEQEND0						Enable or disable interrupt for SEQEND[0] event
								See <a href="#">EVENTS_SEQEND[0]</a>
			Disabled				0	Disable
			Enabled				1	Enable
F	RW	SEQEND1						Enable or disable interrupt for SEQEND[1] event

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	H G F E D C B				
<b>Reset 0x00000000</b>	0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
					See <a href="#">EVENTS_SEQEND[1]</a>
			Disabled	0	Disable
			Enabled	1	Enable
G RW PWMPERIODEND					Enable or disable interrupt for PWMPERIODEND event
					See <a href="#">EVENTS_PWMPERIODEND</a>
			Disabled	0	Disable
			Enabled	1	Enable
H RW LOOPSDONE					Enable or disable interrupt for LOOPSDONE event
					See <a href="#">EVENTS_LOOPSDONE</a>
			Disabled	0	Disable
			Enabled	1	Enable

### 47.5.3 INTSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	H G F E D C B				
<b>Reset 0x00000000</b>	0 0				
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>
B RW STOPPED					Write '1' to Enable interrupt for STOPPED event
					See <a href="#">EVENTS_STOPPED</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C RW SEQSTARTED0					Write '1' to Enable interrupt for SEQSTARTED[0] event
					See <a href="#">EVENTS_SEQSTARTED[0]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D RW SEQSTARTED1					Write '1' to Enable interrupt for SEQSTARTED[1] event
					See <a href="#">EVENTS_SEQSTARTED[1]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E RW SEQEND0					Write '1' to Enable interrupt for SEQEND[0] event
					See <a href="#">EVENTS_SEQEND[0]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F RW SEQEND1					Write '1' to Enable interrupt for SEQEND[1] event
					See <a href="#">EVENTS_SEQEND[1]</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G RW PWMPERIODEND					Write '1' to Enable interrupt for PWMPERIODEND event
					See <a href="#">EVENTS_PWMPERIODEND</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H RW LOOPSDONE					Write '1' to Enable interrupt for LOOPSDONE event

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	H G F E D C B				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_LOOPSDONE</a>
		Set	1	Enable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

## 47.5.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	H G F E D C B				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
B	RW	STOPPED			Write '1' to Disable interrupt for STOPPED event
					See <a href="#">EVENTS_STOPPED</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
C	RW	SEQSTARTED0			Write '1' to Disable interrupt for SEQSTARTED[0] event
					See <a href="#">EVENTS_SEQSTARTED[0]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
D	RW	SEQSTARTED1			Write '1' to Disable interrupt for SEQSTARTED[1] event
					See <a href="#">EVENTS_SEQSTARTED[1]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
E	RW	SEQEND0			Write '1' to Disable interrupt for SEQEND[0] event
					See <a href="#">EVENTS_SEQEND[0]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
F	RW	SEQEND1			Write '1' to Disable interrupt for SEQEND[1] event
					See <a href="#">EVENTS_SEQEND[1]</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
G	RW	PWMPERIODEND			Write '1' to Disable interrupt for PWMPERIODEND event
					See <a href="#">EVENTS_PWMPERIODEND</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	
H	RW	LOOPSDONE			Write '1' to Disable interrupt for LOOPSDONE event
					See <a href="#">EVENTS_LOOPSDONE</a>
		Clear	1	Disable	
		Disabled	0	Read: Disabled	
		Enabled	1	Read: Enabled	

## 47.5.5 ENABLE

Address offset: 0x500

## PWM module enable register

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

## 47.5.6 MODE

Address offset: 0x504

Selects operating mode of the wave counter

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A																															
<b>Reset 0x00000000</b>																																
Id	RW	Field	Value Id	Value	Description																											
A	RW	UPDOWN			Selects up or up and down as wave counter mode																											
		Up	0		Up counter - edge aligned PWM duty-cycle																											
		UpAndDown	1		Up and down counter - center aligned PWM duty cycle																											

## **47.5.7 COUNTERTOP**

Address offset: 0x508

Value up to which the pulse generator counter counts

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x000003FF</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
Id	RW	Field	Value Id	Value	Description																											
A	RW	COUNTERTOP	[3..32767]		Value up to which the pulse generator counter counts. This register is ignored when DECODER.MODE=WaveForm and only values from RAM will be used.																											

## 47.5.8 PRESCALER

Address offset: 0x50C

## Configuration for PWM\_CLK

## 47.5.9 DECODER

Address offset: 0x510

## Configuration of the decoder

## **47.5.10 LOOP**

Address offset: 0x514

## Amount of playback of a loop

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	CNT			Amount of playback of pattern cycles																											
		Disabled	0	Looping disabled (stop at the end of the sequence)																												

## 47.5.11 SEQ[0].PTR

Address offset: 0x520

Beginning address in Data RAM of this sequence

## 47.5.12 SEQ[0].CNT

Address offset: 0x524

Amount of values (duty cycles) in this sequence

## 47.5.13 SEQ[0].REFRESH

Address offset: 0x528

Amount of additional PWM periods between samples loaded into compare register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000001</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	CNT				Amount of additional PWM periods between samples loaded into compare register (load every REFRESH.CNT+1 PWM periods)																										
		Continuous	0			Update every PWM period																										

#### **47.5.14 SEQ[0].ENDDELAY**

Address offset: 0x52C

Time added after the sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id												A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	CNT			Time added after the sequence in PWM periods																											

## 47.5.15 SEQ[1].PTR

Address offset: 0x540

Beginning address in Data RAM of this sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value		Description																										
A	RW	PTR				Beginning address in Data RAM of this sequence																										

## 47.5.16 SEQ[1].CNT

Address offset: 0x544

Amount of values (duty cycles) in this sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Id																																	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																												
A	RW	CNT			Amount of values (duty cycles) in this sequence																												
		Disabled	0		Sequence is disabled, and shall not be started as it is empty																												

## 47.5.17 SEQ[1].REFRESH

Address offset: 0x548

Amount of additional PWM periods between samples loaded into compare register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	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	1	
Id	RW	Field	Value Id	Value		Description																										
A	RW	CNT				Amount of additional PWM periods between samples loaded into compare register (load every REFRESH.CNT+1 PWM periods)																										
		Continuous	0			Update every PWM period																										

## 47.5.18 SEQ[1].ENDDELAY

Address offset: 0x54C

## Time added after the sequence

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id	Value	Description																											
A	RW	CNT			Time added after the sequence in PWM periods																											

## 47.5.19 PSEL.OUT[0]

Address offset: 0x560

### Output pin select for PWM channel 0

## 47.5.20 PSEL.OUT[1]

Address offset: 0x564

## Output pin select for PWM channel 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	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

## 47.5.21 PSEL.OUT[2]

Address offset: 0x568

### Output pin select for PWM channel 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	B																											A	A	A	A	A
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Id	RW	Field	Value Id	Value	Description																											
A	RW	PIN		[0..31]	Pin number																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

## 47.5.22 PSEL.OUT[3]

Address offset: 0x56C

### Output pin select for PWM channel 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	B																											A	A	A	A	A
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>	<b>Description</b>																											
B	RW	CONNECT			Connection																											
			Disconnected	1	Disconnect																											
			Connected	0	Connect																											

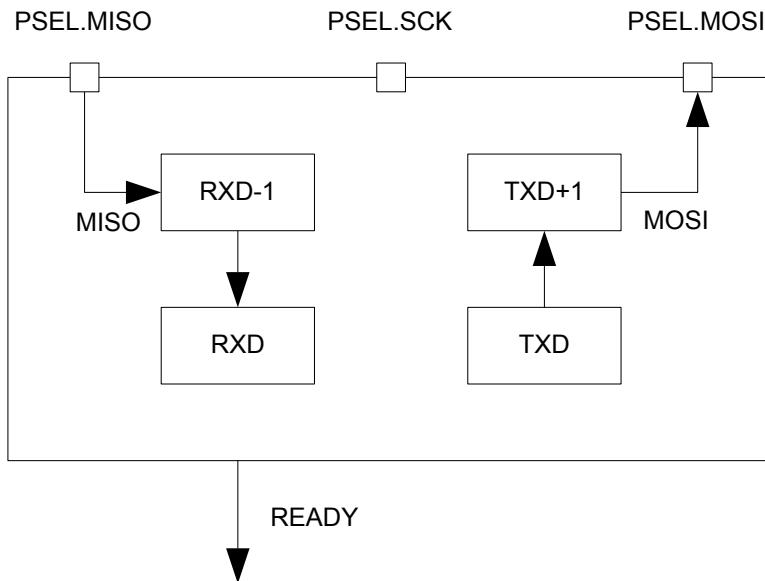
## **47.6 Electrical specification**

#### **47.6.1 PWM Electrical Specification**

Symbol	Description	Min.	Typ.	Max.	Units
$I_{PWM,16MHz}$	PWM run current, Prescaler set to DIV_1 (16 MHz), excluding DMA and GPIO		200		µA
$I_{PWM,8MHz}$	PWM run current, Prescaler set to DIV_2 (8 MHz), excluding DMA and GPIO		150		µA
$I_{PWM,125kHz}$	PWM run current, Prescaler set to DIV_128 (125 kHz), excluding DMA and GPIO		150		µA

## 48 SPI — Serial peripheral interface master

The SPI master provides a simple CPU interface which includes a TXD register for sending data and an RXD register for receiving data. This section is added for legacy support for now.



**Figure 150: SPI master**

RXD-1 and TXD+1 illustrate the double buffered version of RXD and TXD respectively.

### 48.1 Functional description

The TXD and RXD registers are double-buffered to enable some degree of uninterrupted data flow in and out of the SPI master.

The SPI master does not implement support for chip select directly. Therefore, the CPU must use available GPIOs to select the correct slave and control this independently of the SPI master. The SPI master supports SPI modes 0 through 3.

**Table 120: SPI modes**

Mode	Clock polarity CPOL	Clock phase CPHA
SPI_MODE 0 (Leading)	0 (Active High)	
SPI_MODE 0 (Leading)	1 (Active Low)	
SPI_MODE 1 (Trailing)	0 (Active High)	
SPI_MODE 1 (Trailing)	1 (Active Low)	

#### 48.1.1 SPI master mode pin configuration

The different signals SCK, MOSI, and MISO associated with the SPI master are mapped to physical pins.

This mapping is according to the configuration specified in the PSEL\_SCK, PSEL\_MOSI, and PSEL\_MISO registers respectively. If a value of 0xFFFFFFFF is specified in any of these registers, the associated SPI master signal is not connected to any physical pin. The PSEL\_SCK, PSEL\_MOSI, and PSEL\_MISO registers and their configurations are only used as long as the SPI master is enabled, and retained only as long as the device is in ON mode. PSEL\_SCK, PSEL\_MOSI, and PSEL\_MISO must only be configured when the SPI master is disabled.

To secure correct behavior in the SPI, the pins used by the SPI must be configured in the GPIO peripheral as described in [Table 121: GPIO configuration](#) on page 514 prior to enabling the SPI. The SCK must

always be connected to a pin, and that pin's input buffer must always be connected for the SPI to work. This configuration must be retained in the GPIO for the selected IOs as long as the SPI is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time, failing to do so may result in unpredictable behavior.

**Table 121: GPIO configuration**

SPI master signal	SPI master pin	Direction	Output value
SCK	As specified in PSEL_SCK	Output	Same as CONFIG.CPOL
MOSI	As specified in PSEL_MOSI	Output	0
MISO	As specified in PSEL_MISO	Input	Not applicable

### 48.1.2 Shared resources

The SPI shares registers and other resources with other peripherals that have the same ID as the SPI. Therefore, the user must disable all peripherals that have the same ID as the SPI before the SPI can be configured and used.

Disabling a peripheral that has the same ID as the SPI will not reset any of the registers that are shared with the SPI. It is therefore important to configure all relevant SPI registers explicitly to secure that it operates correctly.

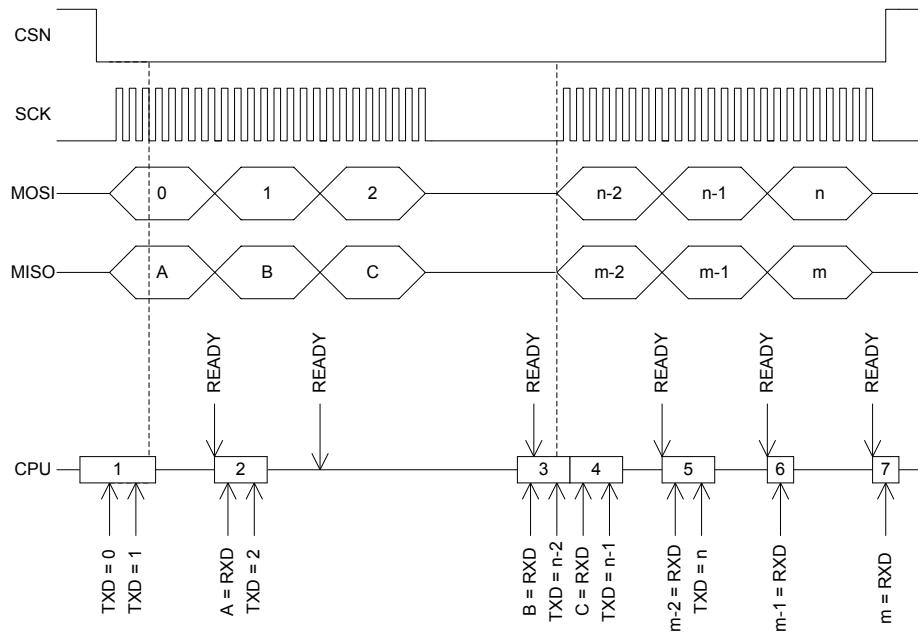
See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

### 48.1.3 SPI master transaction sequence

An SPI master transaction is started by writing the first byte, which is to be transmitted by the SPI master, to the TXD register.

Since the transmitter is double buffered, the second byte can be written to the TXD register immediately after the first one. The SPI master will then send these bytes in the order they are written to the TXD register.

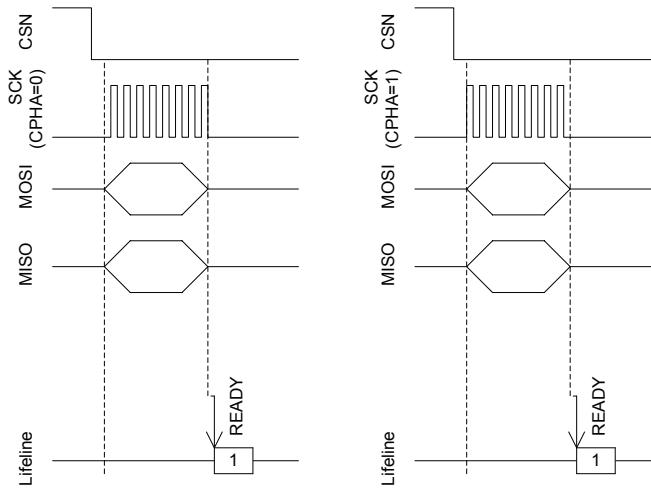
The SPI master is a synchronous interface, and for every byte that is sent, a different byte will be received at the same time; this is illustrated in [Figure 151: SPI master transaction](#) on page 515. Bytes that are received will be moved to the RXD register where the CPU can extract them by reading the register. The RXD register is double buffered in the same way as the TXD register, and a second byte can therefore be received at the same time as the first byte is being extracted from RXD by the CPU. The SPI master will generate a READY event every time a new byte is moved to the RXD register. The double buffered byte will be moved from RXD-1 to RXD as soon as the first byte is extracted from RXD. The SPI master will stop when there are no more bytes to send in TXD and TXD+1.



**Figure 151: SPI master transaction**

The READY event of the third byte transaction is delayed until B is extracted from RXD in occurrence number 3 on the horizontal lifeline. The reason for this is that the third event is generated first when C is moved from RXD-1 to RXD after B is read.

The SPI master will move the incoming byte to the RXD register after a short delay following the SCK clock period of the last bit in the byte. This also means that the READY event will be delayed accordingly, see [Figure 152: SPI master transaction](#) on page 515. Therefore, it is important that you always clear the READY event, even if the RXD register and the data that is being received is not used.



**Figure 152: SPI master transaction**

## 48.2 Registers

**Table 122: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	SPI	SPI0	SPI master 0	Deprecated
0x40004000	SPI	SPI1	SPI master 1	Deprecated
0x40023000	SPI	SPI2	SPI master 2	Deprecated

**Table 123: Register Overview**

Register	Offset	Description
EVENTS_READY	0x108	TXD byte sent and RXD byte received
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>ENABLE</i>	0x500	Enable SPI
<i>PSEL.SCK</i>	0x508	Pin select for SCK
<i>PSEL.MOSI</i>	0x50C	Pin select for MOSI
<i>PSEL.MISO</i>	0x510	Pin select for MISO
<i>PSEL.SCK</i>	0x508	Pin select for SCK
<i>PSEL.MOSI</i>	0x50C	Pin select for MOSI
<i>PSEL.MISO</i>	0x510	Pin select for MISO
<i>RXD</i>	0x518	RXD register
<i>TXD</i>	0x51C	TXD register
<i>FREQUENCY</i>	0x524	SPI frequency
<i>CONFIG</i>	0x554	Configuration register

### 48.2.1 INTENSET

Address offset: 0x304

Enable interrupt

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Id																																				
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Id	RW	Field	Value	Id	Value	Description																														
A	RW	READY				Write '1' to Enable interrupt for READY event																														
						See <a href="#">EVENTS_READY</a>																														
		Set	1			Enable																														
		Disabled	0			Read: Disabled																														
		Enabled	1			Read: Enabled																														

### 48.2.2 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id																																			
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Id	RW	Field	Value	Id	Value	Description																													
A	RW	READY				Write '1' to Disable interrupt for READY event																													
						See <a href="#">EVENTS_READY</a>																													
		Clear	1			Disable																													
		Disabled	0			Read: Disabled																													
		Enabled	1			Read: Enabled																													

### **48.2.3 ENABLE**

Address offset: 0x500

## Enable SPI

#### **48.2.4 PSEL\_SCK ( Deprecated )**

Address offset: 0x508

## Pin select for SCK

#### **48.2.5 PSELMOSSI ( Deprecated )**

Address offset: 0x50C

## Pin select for MOSI

## **48.2.6 PSELMISO ( Deprecated )**

Address offset: 0x510

## Pin select for MISO

## 48.2.7 PSEL.SCK

Address offset: 0x508

## Pin select for SCK

## 48.2.8 PSEL.MOSI

Address offset: 0x50C

## Pin select for MOSI

## 48.2.9 PSEL.MISO

Address offset: 0x510

## Pin select for MISO

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset OxFFFFFFF</b>	1	1	1	1	1	1	1	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	PSEL MISO	[0..31]		Pin number configuration for SPI MISO signal																											
		Disconnected	0xFFFFFFFF		Disconnect																											

48.2.10 RXD

Address offset: 0x518

RXD register

#### **48.2.11 TXD**

Address offset: 0x51C

## TXD register

## 48.2.12 FREQUENCY

Address offset: 0x524

## SPI frequency

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
<b>Reset 0x04000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>																									
Id	RW	Field	Value Id	Value		Description																										
A	RW	FREQUENCY				SPI master data rate																										
		K125		0x02000000		125 kbps																										
		K250		0x04000000		250 kbps																										
		K500		0x08000000		500 kbps																										
		M1		0x10000000		1 Mbps																										

## 48.2.13 CONFIG

Address offset: 0x554

## Configuration register

### **48.3 Electrical specification**

### 48.3.1 SPI master interface

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{SPI}}$	Bit rates for SPI <sup>39</sup>			8 <sup>40</sup>	Mbps
$I_{\text{SPI},2\text{Mbps}}$	Run current for SPI, 2 Mbps			50	$\mu\text{A}$
$I_{\text{SPI},8\text{Mbps}}$	Run current for SPI, 8 Mbps			50	$\mu\text{A}$
$I_{\text{SPI,IDL}}$	Idle current for SPI (STARTed, no CSN activity)		<1		$\mu\text{A}$
$t_{\text{SPI,START,LP}}$	Time from writing TXD register to transmission started, low power mode		$t_{\text{SPI,START,CL}}$		$\mu\text{s}$
			+		
				$t_{\text{START\_HFIN}}$	
$t_{\text{SPI,START,CL}}$	Time from writing TXD register to transmission started, constant latency mode		1		$\mu\text{s}$

#### 48.3.2 Serial Peripheral Interface (SPI) Master timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPI,CSCK,8Mbps}$	SCK period at 8Mbps		125		ns
$t_{SPI,CSCK,4Mbps}$	SCK period at 4Mbps		250		ns
$t_{SPI,CSCK,2Mbps}$	SCK period at 2Mbps		500		ns
$t_{SPI,RSCK,LD}$	SCK rise time, low drive <sup>a</sup>			$t_{RF,25pF}$	
$t_{SPI,RSCK,HD}$	SCK rise time, high drive <sup>a</sup>			$t_{HRF,25pF}$	
$t_{SPI,FSCK,LD}$	SCK fall time, low drive <sup>a</sup>			$t_{RF,25pF}$	
$t_{SPI,FSCK,HD}$	SCK fall time, high drive <sup>a</sup>			$t_{HRF,25pF}$	

<sup>39</sup> Higher bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

<sup>40</sup> The actual maximum data rate depends on the slave's CLK to MISO and MOSI setup and hold timings.

<sup>a</sup> At 25pF load, including GPIO capacitance, see GPIO spec.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPI,WHSCK}$	SCK high time <sup>a</sup>			$(0.5*t_{CSCK})$	
				$-t_{RSCK}$	
$t_{SPI,WLSCK}$	SCK low time <sup>a</sup>			$(0.5*t_{CSCK})$	
				$-t_{FSCK}$	
$t_{SPI,SUMI}$	MISO to CLK edge setup time	19			ns
$t_{SPI,HMI}$	CLK edge to MISO hold time	18			ns
$t_{SPI,VMO}$	CLK edge to MOSI valid			59	ns
$t_{SPI,HMO}$	MOSI hold time after CLK edge	20			ns

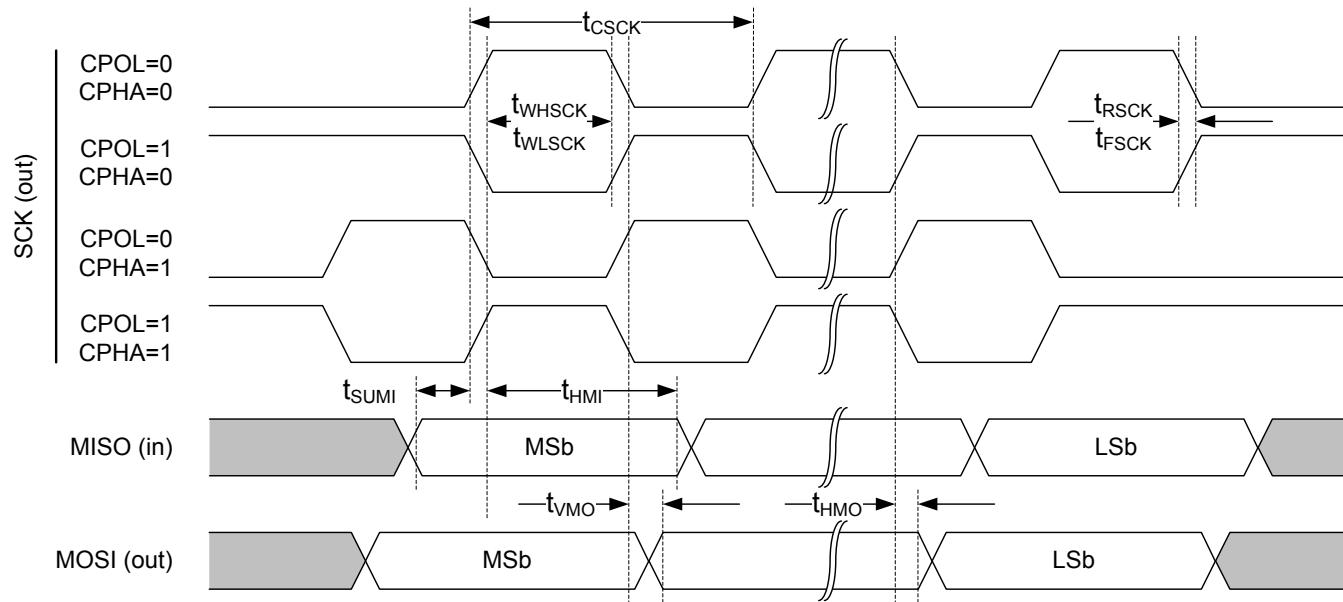


Figure 153: SPI master timing diagram

# 49 TWI — I<sup>2</sup>C compatible two-wire interface

The TWI master is compatible with I<sup>2</sup>C operating at 100 kHz and 400 kHz.

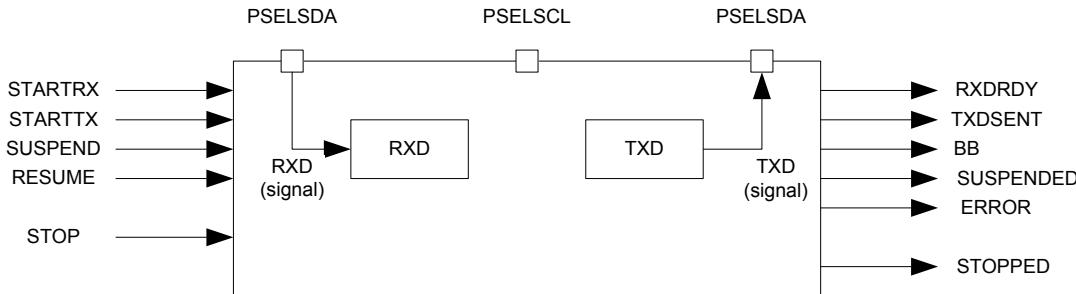


Figure 154: TWI master's main features

## 49.1 Functional description

This TWI master is not compatible with CBUS. The TWI transmitter and receiver are single buffered.

See, [Figure 154: TWI master's main features](#) on page 521.

A TWI setup comprising one master and three slaves is illustrated in [Figure 155: A typical TWI setup comprising one master and three slaves](#) on page 521. This TWI master is only able to operate as the only master on the TWI bus.

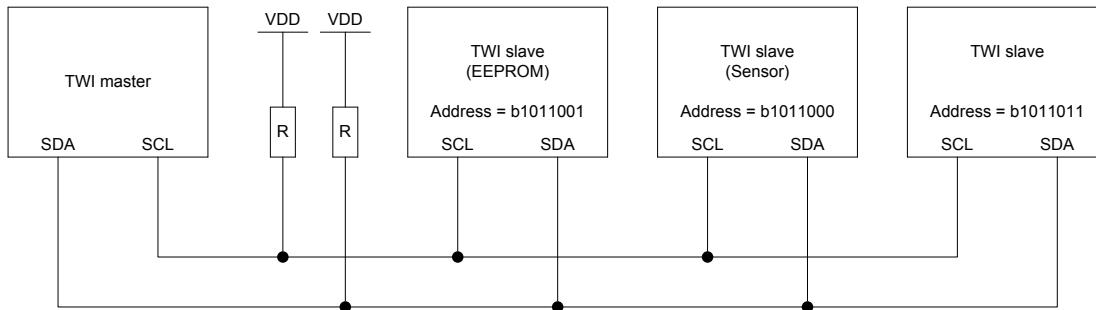


Figure 155: A typical TWI setup comprising one master and three slaves

This TWI master supports clock stretching performed by the slaves. The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

## 49.2 Master mode pin configuration

The different signals SCL and SDA associated with the TWI master are mapped to physical pins according to the configuration specified in the PSELSCl and PSELSDA registers respectively.

If a value of 0xFFFFFFFF is specified in any of these registers, the associated TWI master signal is not connected to any physical pin. The PSELSCl and PSELSDA registers and their configurations are only used

as long as the TWI master is enabled, and retained only as long as the device is in ON mode. PSEL\_SCL and PSEL\_SDA must only be configured when the TWI is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in [Table 124: GPIO configuration](#) on page 522.

Only one peripheral can be assigned to drive a particular GPIO pin at a time, failing to do so may result in unpredictable behavior.

**Table 124: GPIO configuration**

TWI master signal	TWI master pin	Direction	Drive strength	Output value
SCL	As specified in PSEL_SCL	Input	S0D1	Not applicable
SDA	As specified in PSEL_SDA	Input	S0D1	Not applicable

## 49.3 Shared resources

The TWI shares registers and other resources with other peripherals that have the same ID as the TWI.

Therefore, you must disable all peripherals that have the same ID as the TWI before the TWI can be configured and used. Disabling a peripheral that has the same ID as the TWI will not reset any of the registers that are shared with the TWI. It is therefore important to configure all relevant TWI registers explicitly to secure that it operates correctly.

The Instantiation table in [Instantiation](#) on page 24 shows which peripherals have the same ID as the TWI.

## 49.4 Master write sequence

A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

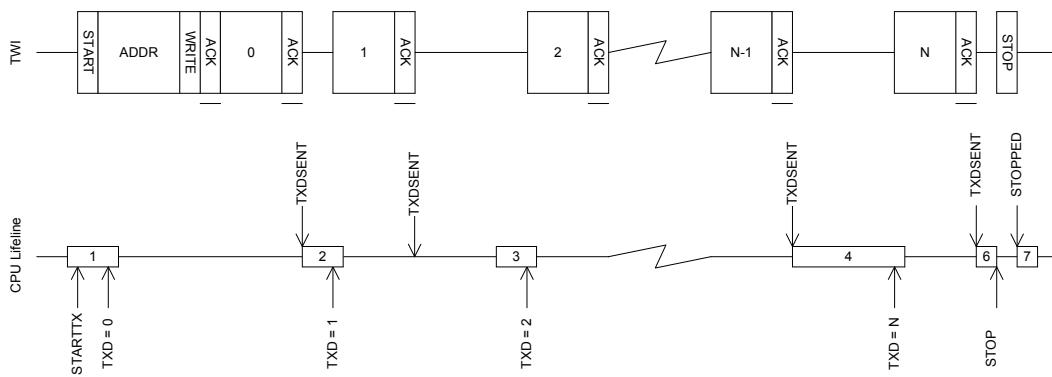
The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes that are written to the TXD register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave. A TXDSENT event will be generated each time the TWI master has clocked out a TXD byte, and the associated ACK/NACK bit has been clocked in from the slave.

The TWI master transmitter is single buffered, and a second byte can only be written to the TXD register after the previous byte has been clocked out and the ACK/NACK bit clocked in, that is, after the TXDSENT event has been generated.

If the CPU is prevented from writing to TXD when the TWI master is ready to clock out a byte, the TWI master will stretch the clock until the CPU has written a byte to the TXD register.

A typical TWI master write sequence is illustrated in [Figure 156: The TWI master writing data to a slave](#) on page 523. Occurrence 3 in the figure illustrates delayed processing of the TXDSENT event associated with TXD byte 1. In this scenario the TWI master will stretch the clock to prevent writing erroneous data to the slave.



**Figure 156: The TWI master writing data to a slave**

The TWI master write sequence is stopped when the STOP task is triggered whereupon the TWI master will generate a stop condition on the TWI bus.

## 49.5 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1).

The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

The TWI master will generate a RXDRDY event every time a new byte is received in the RXD register.

After receiving a byte, the TWI master will delay sending the ACK/NACK bit by stretching the clock until the CPU has extracted the received byte, that is, by reading the RXD register.

The TWI master read sequence is stopped by triggering the STOP task. This task must be triggered before the last byte is extracted from RXD to ensure that the TWI master sends a NACK back to the slave before generating the stop condition.

A typical TWI master read sequence is illustrated in [Figure 157: The TWI master reading data from a slave](#) on page 524. Occurrence 3 in this figure illustrates delayed processing of the RXDRDY event associated with RXD byte B. In this scenario the TWI master will stretch the clock to prevent the slave from overwriting the contents of the RXD register.

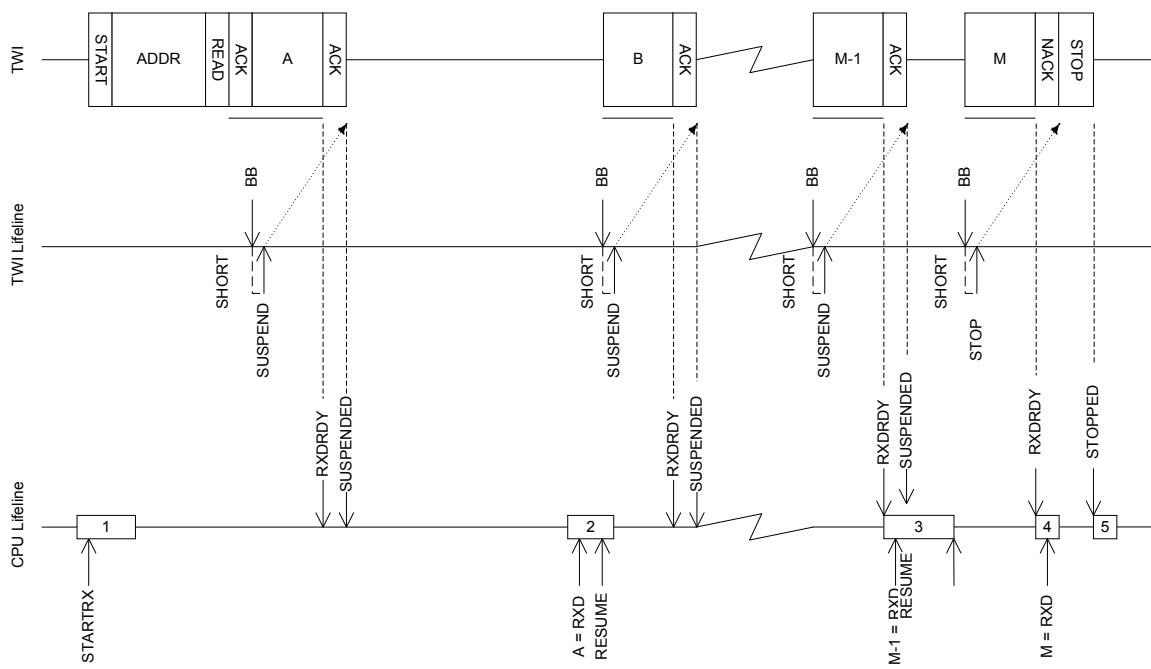


Figure 157: The TWI master reading data from a slave

## 49.6 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes one byte to the slave followed by reading M bytes from the slave. Any combination and number of transmit and receive sequences can be combined in this fashion. Only one shortcut to STOP can be enabled at any given time.

The figure below illustrates a repeated start sequence where the TWI master writes one byte, followed by reading M bytes from the slave without performing a stop in-between.

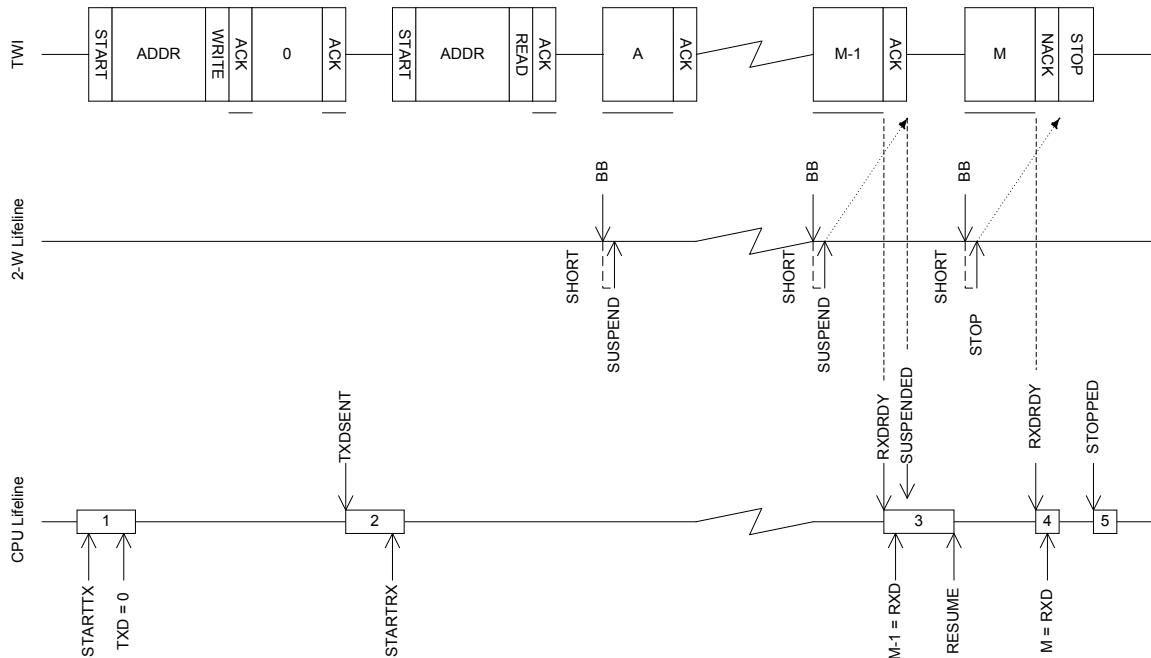


Figure 158: A repeated start sequence, where the TWI master writes one byte, followed by reading M bytes from the slave without performing a stop in-between

To generate a repeated start after a read sequence, a second start task must be triggered instead of the STOP task, that is, STARTRX or STARTTX. This start task must be triggered before the last byte is extracted from RXD to ensure that the TWI master sends a NACK back to the slave before generating the repeated start condition.

## 49.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

## 49.8 Registers

**Table 125: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40003000	TWI	TWI0	Two-wire interface master 0	Deprecated
0x40004000	TWI	TWI1	Two-wire interface master 1	Deprecated

**Table 126: Register Overview**

Register	Offset	Description
TASKS_STARTRX	0x000	Start TWI receive sequence
TASKS_STARTTX	0x008	Start TWI transmit sequence
TASKS_STOP	0x014	Stop TWI transaction
TASKS_SUSPEND	0x01C	Suspend TWI transaction
TASKS_RESUME	0x020	Resume TWI transaction
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_RXDREADY	0x108	TWI RXD byte received
EVENTS_TXDSENT	0x11C	TWI TXD byte sent
EVENTS_ERROR	0x124	TWI error
EVENTS_BB	0x138	TWI byte boundary, generated before each byte that is sent or received
EVENTS_SUSPENDED	0x148	TWI entered the suspended state
<i>SHORTS</i>	0x200	Shortcut register
<i>INTENSET</i>	0x304	Enable interrupt
<i>INTENCLR</i>	0x308	Disable interrupt
<i>ERRORSRC</i>	0x4C4	Error source
<i>ENABLE</i>	0x500	Enable TWI
<i>PSEL SCL</i>	0x508	Pin select for SCL
<i>PSEL SDA</i>	0x50C	Pin select for SDA
<i>RXD</i>	0x518	RXD register
<i>TXD</i>	0x51C	TXD register
<i>FREQUENCY</i>	0x524	TWI frequency
<i>ADDRESS</i>	0x588	Address used in the TWI transfer

### 49.8.1 SHORTS

Address offset: 0x200

Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Id																																		
Reset	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id																															
A	RW	BB_SUSPEND	Value																												Description			
Shortcut between BB event and SUSPEND task																																		

## **49.8.2 INTENSET**

Address offset: 0x304

## Enable interrupt

### 49.8.3 INTENCLR

Address offset: 0x308

Disable interrupt

#### **49.8.4 ERRORSRC**

Address offset: 0x4C4

### Error source

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	C	B	A																													
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
			Present	1	Read: error present																											
			Clear	1	Write: clear error on writing '1'																											

## **49.8.5 ENABLE**

Address offset: 0x500

## Enable TWI

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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		0				Enable or disable TWI																								
		Disabled		0				Disable TWI																								
		Enabled		5				Enable TWI																								

## 49.8.6 PSEL SCL

Address offset: 0x508

## Pin select for SCL

## 49.8.7 PSELSDA

Address offset: 0x50C

## Pin select for SDA

49.8.8 RXD

Address offset: 0x518

## RXD register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	R	RXD			RXD register																											

49.8.9 TXD

Address offset: 0x51C

## TXD register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																										A	A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value	Description																											
A	RW	TXD			TXD register																											

## 49.8.10 FREQUENCY

Address offset: 0x524

## TWI frequency

## **49.8.11 ADDRESS**

Address offset: 0x588

Address used in the TWI transfer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																											A	A	A	A	A	A
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value																		Description										
A	RW	ADDRESS																				Address used in the TWI transfer										

## **49.9 Electrical specification**

### **49.9.1 TWI interface electrical specifications**

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{TWI}}$	Bit rates for TWI <sup>41</sup>	100	400	kbps	
$I_{\text{TWI},100\text{ kbps}}$	Run current for TWI, 100 kbps		50		μA
$I_{\text{TWI},400\text{ kbps}}$	Run current for TWI, 400 kbps		50		μA
$t_{\text{TWI,START,LP}}$	Time from STARTRX/STARTTX task to transmission started, Low power mode		$t_{\text{TWI,START,C}}$ + $t_{\text{START\_HFIN}}$		μs
$t_{\text{TWI,START,CL}}$	Time from STARTRX/STARTTX task to transmission started, Constant latency mode		1.5		μs

#### **49.9.2 Two Wire Interface (TWI) timing specifications**

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWI,SCL,100kbps}$	SCL clock frequency, 100 kbps		100		kHz
$f_{TWI,SCL,250kbps}$	SCL clock frequency, 250 kbps		250		kHz
$f_{TWI,SCL,400kbps}$	SCL clock frequency, 400 kbps		400		kHz
$t_{TWI,SU_DAT}$	Data setup time before positive edge on SCL – all modes	300			ns
$t_{TWI,HD_DAT}$	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWI,HD_STA,100kbps}$	TWI master hold time for START and repeated START condition, 100 kbps	10000			ns

---

<sup>41</sup> Higher bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TWI,HD\_STA,250kbps}$	TWI master hold time for START and repeated START condition, 250kbps	4000			ns
$t_{TWI,HD\_STA,400kbps}$	TWI master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWI,SU\_STO,100kbps}$	TWI master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWI,SU\_STO,250kbps}$	TWI master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWI,SU\_STO,400kbps}$	TWI master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWI,BUF,100kbps}$	TWI master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWI,BUF,250kbps}$	TWI master bus free time between STOP and START conditions, 250 kbps	2700			ns
$t_{TWI,BUF,400kbps}$	TWI master bus free time between STOP and START conditions, 400 kbps	2100			ns

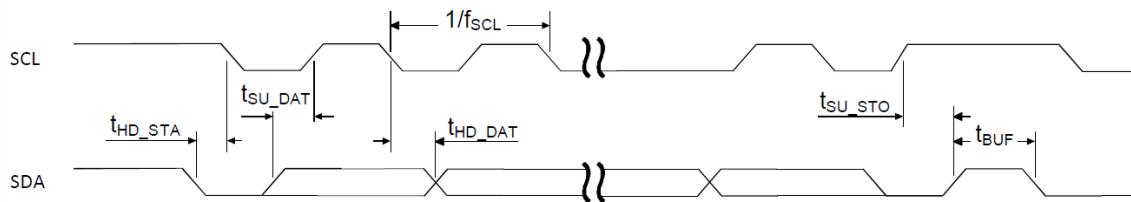


Figure 159: TWI timing diagram, 1 byte transaction

# 50 UART — Universal asynchronous receiver/ transmitter

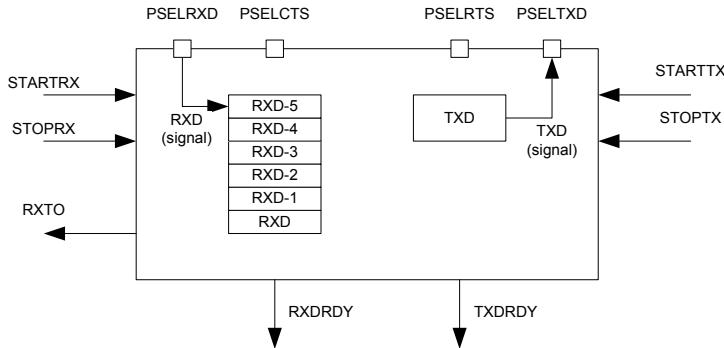


Figure 160: UART configuration

## 50.1 Functional description

Listed here are the main features of UART.

The UART implements support for the following features:

- Full-duplex operation
- Automatic flow control
- Parity checking and generation for the 9<sup>th</sup> data bit

As illustrated in [Figure 160: UART configuration](#) on page 531, the UART uses the TXD and RXD registers directly to transmit and receive data. The UART uses one stop bit.

## 50.2 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UART are mapped to physical pins according to the configuration specified in the PSELRXD, PSELCTS, PSELRTS, and PSELTXD registers respectively.

If a value of 0xFFFFFFFF is specified in any of these registers, the associated UART signal will not be connected to any physical pin. The PSELRXD, PSELCTS, PSELRTS, and PSELTXD registers and their configurations are only used as long as the UART is enabled, and retained only for the duration the device is in ON mode. PSELRXD, PSELCTS, PSELRTS and PSELTXD must only be configured when the UART is disabled.

To secure correct signal levels on the pins by the UART when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in [Pin configuration](#) on page 531.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

Table 127: GPIO configuration

UART pin	Direction	Output value
RXD	Input	Not applicable
CTS	Input	Not applicable
RTS	Output	1
TXD	Output	1

## 50.3 Shared resources

The UART shares registers and other resources with other peripherals that have the same ID as the UART.

Therefore, you must disable all peripherals that have the same ID as the UART before the UART can be configured and used. Disabling a peripheral that has the same ID as the UART will not reset any of the registers that are shared with the UART. It is therefore important to configure all relevant UART registers explicitly to ensure that it operates correctly.

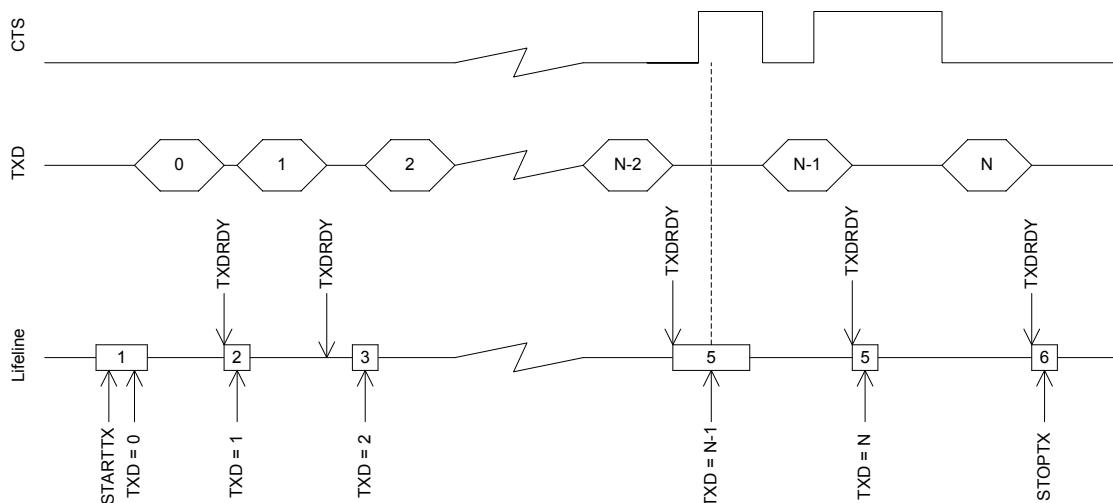
See the Instantiation table in [Instantiation](#) on page 24 for details on peripherals and their IDs.

## 50.4 Transmission

A UART transmission sequence is started by triggering the STARTTX task.

Bytes are transmitted by writing to the TXD register. When a byte has been successfully transmitted the UART will generate a TXDRDY event after which a new byte can be written to the TXD register. A UART transmission sequence is stopped immediately by triggering the STOPTX task.

If flow control is enabled a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in [Figure 161: UART transmission](#) on page 532. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended. For more information, see [Suspending the UART](#) on page 533.



**Figure 161: UART transmission**

## 50.5 Reception

A UART reception sequence is started by triggering the STARTRX task.

The UART receiver chain implements a FIFO capable of storing six incoming RXD bytes before data is overwritten. Bytes are extracted from this FIFO by reading the RXD register. When a byte is extracted from the FIFO a new byte pending in the FIFO will be moved to the RXD register. The UART will generate an RXDRDY event every time a new byte is moved to the RXD register.

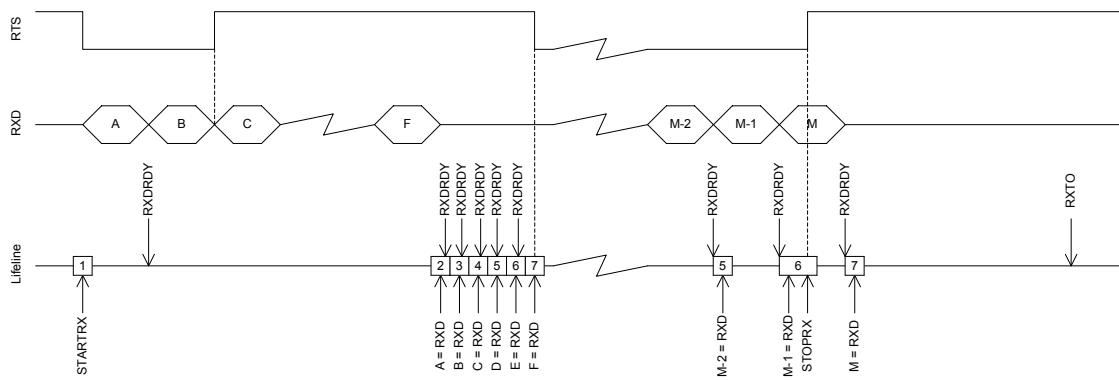
When flow control is enabled, the UART will deactivate the RTS signal when there is only space for four more bytes in the receiver FIFO. The counterpart transmitter is therefore able to send up to four bytes after the RTS signal is deactivated before data is being overwritten. To prevent overwriting data in the FIFO, the counterpart UART transmitter must therefore make sure to stop transmitting data within four bytes after the RTS line is deactivated.

The RTS signal will first be activated again when the FIFO has been emptied, that is, when all bytes in the FIFO have been read by the CPU, see [Figure 162: UART reception](#) on page 533.

The RTS signal will also be deactivated when the receiver is stopped through the STOPRX task as illustrated in [Figure 162: UART reception](#) on page 533. The UART is able to receive four to five additional bytes if they are sent in succession immediately after the RTS signal has been deactivated. This is possible because the UART is, even after the STOPRX task is triggered, able to receive bytes for an extended period of time dependent on the configured baud rate. The UART will generate a receiver timeout event (RXTO) when this period has elapsed.

To prevent loss of incoming data the RXD register must only be read one time following every RXDRDY event.

To secure that the CPU can detect all incoming RXDRDY events through the RXDRDY event register, the RXDRDY event register must be cleared before the RXD register is read. The reason for this is that the UART is allowed to write a new byte to the RXD register, and therefore can also generate a new event, immediately after the RXD register is read (emptied) by the CPU.



**Figure 162: UART reception**

As indicated in occurrence 2 in the figure, the RXDRDY event associated with byte B is generated first after byte A has been extracted from RXD.

## 50.6 Suspending the UART

The UART can be suspended by triggering the SUSPEND task.

SUSPEND will affect both the UART receiver and the UART transmitter, i.e. the transmitter will stop transmitting and the receiver will stop receiving. UART transmission and reception can be resumed, after being suspended, by triggering STARTTX and STARTRX respectively.

Following a SUSPEND task, an ongoing TXD byte transmission will be completed before the UART is suspended.

When the SUSPEND task is triggered, the UART receiver will behave in the same way as it does when the STOPRX task is triggered.

## 50.7 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

## 50.8 Using the UART without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

## 50.9 Parity configuration

When parity is enabled, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

## 50.10 Registers

**Table 128: Instances**

Base address	Peripheral	Instance	Description	Configuration
0x40002000	UART	UART0	Universal Asynchronous Receiver/ Transmitter	Deprecated

**Table 129: Register Overview**

Register	Offset	Description
TASKS_STARTRX	0x000	Start UART receiver
TASKS_STOPRX	0x004	Stop UART receiver
TASKS_STARTTX	0x008	Start UART transmitter
TASKS_STOPTX	0x00C	Stop UART transmitter
TASKS_SUSPEND	0x01C	Suspend UART
EVENTS_CTS	0x100	CTS is activated (set low). Clear To Send.
EVENTS_NCTS	0x104	CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108	Data received in RXD
EVENTS_TXDRDY	0x11C	Data sent from TXD
EVENTS_ERROR	0x124	Error detected
EVENTS_RXTO	0x144	Receiver timeout
<b>SHORTS</b>	0x200	Shortcut register
<b>INTENSET</b>	0x304	Enable interrupt
<b>INTENCLR</b>	0x308	Disable interrupt
<b>ERRORSRC</b>	0x480	Error source
<b>ENABLE</b>	0x500	Enable UART
<b>PSELRTS</b>	0x508	Pin select for RTS
<b>PSELTXD</b>	0x50C	Pin select for TXD
<b>PSELCTS</b>	0x510	Pin select for CTS
<b>PSELRXD</b>	0x514	Pin select for RXD
<b>RXD</b>	0x518	RXD register
<b>TXD</b>	0x51C	TXD register
<b>BAUDRATE</b>	0x524	Baud rate
<b>CONFIG</b>	0x56C	Configuration of parity and hardware flow control

## **50.10.1 SHORTS**

Address offset: 0x200

## Shortcut register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																												B	A			
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value	Id	Value	Description																										
A	RW	CTS_STARTRX				Shortcut between CTS event and STARTRX task																										

Bit number	31 30 29 28 27 26 25 24 23 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				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
					See <a href="#">EVENTS_CTS</a> and <a href="#">TASKS_STARTRX</a>
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
<b>B RW NCTS_STOPRX</b>					Shortcut between NCTS event and STOPRX task
					See <a href="#">EVENTS_NCTS</a> and <a href="#">TASKS_STOPRX</a>
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

## 50.10.2 INTENSET

Address offset: 0x304

Enable interrupt

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0				
Id	F E D C B A				
<b>Reset 0x00000000</b>	<b>0 0</b>				
Id	RW	Field	Value Id	Value	Description
A	RW	CTS			Write '1' to Enable interrupt for CTS event
					See <a href="#">EVENTS_CTS</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
<b>B RW NCTS</b>					Write '1' to Enable interrupt for NCTS event
					See <a href="#">EVENTS_NCTS</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
<b>C RW RXDRDY</b>					Write '1' to Enable interrupt for RXDRDY event
					See <a href="#">EVENTS_RXDRDY</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
<b>D RW TXDRDY</b>					Write '1' to Enable interrupt for TXDRDY event
					See <a href="#">EVENTS_TXDRDY</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
<b>E RW ERROR</b>					Write '1' to Enable interrupt for ERROR event
					See <a href="#">EVENTS_ERROR</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
<b>F RW RXTO</b>					Write '1' to Enable interrupt for RXTO event
					See <a href="#">EVENTS_RXTO</a>
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

## 50.10.3 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																							F		E	D		C	B	A		
<b>Reset 0x00000000</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>		
<b>Id</b>	<b>RW</b>	<b>Field</b>	<b>Value Id</b>	<b>Value</b>		<b>Description</b>																										
A	RW	CTS				Write '1' to Disable interrupt for CTS event																										
						See <a href="#">EVENTS_CTS</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
B	RW	NCTS				Write '1' to Disable interrupt for NCTS event																										
						See <a href="#">EVENTS_NCTS</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
C	RW	RXDRDY				Write '1' to Disable interrupt for RXDRDY event																										
						See <a href="#">EVENTS_RXDRDY</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
D	RW	TXDRDY				Write '1' to Disable interrupt for TXDRDY event																										
						See <a href="#">EVENTS_TXDRDY</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
E	RW	ERROR				Write '1' to Disable interrupt for ERROR event																										
						See <a href="#">EVENTS_ERROR</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										
F	RW	RXTO				Write '1' to Disable interrupt for RXTO event																										
						See <a href="#">EVENTS_RXTO</a>																										
		Clear	1			Disable																										
		Disabled	0			Read: Disabled																										
		Enabled	1			Read: Enabled																										

#### 50.10.4 ERRORSRC

Address offset: 0x480

### Error source

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id																												D	C	B	A			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Id	RW	Field	Value Id																										Description					
A RW OVERRUN																												Overrun error						
																												A start bit is received while the previous data still lies in RXD. (Previous data is lost.)						
																												NotPresent						
																												0						
																												Read: error not present						
																												Present						
B RW PARITY																												Parity error						
																												A character with bad parity is received, if HW parity check is enabled.						
																												NotPresent						
																												0						
																												Read: error not present						
																												Present						
C RW FRAMING																												Framing error occurred						

## **50.10.5 ENABLE**

Address offset: 0x500

## Enable UART

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

## **50.10.6 PSELRTS**

Address offset: 0x508

## Pin select for RTS

## 50.10.7 PSELTXD

Address offset: 0x50C

Pin select for TXD

## **50.10.8 PSELCTS**

Address offset: 0x510

## Pin select for CTS

## **50.10.9 PSELRXD**

Address offset: 0x514

## Pin select for RXD

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
<b>Reset 0xFFFFFFFF</b>	1	1	1	1	1	1	1	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	PSELRXD		[0..31]	Pin number configuration for UART RXD signal																											
			Disconnected	0xFFFFFFFF	Disconnect																											

50.10.10 RXD

Address offset: 0x518

## RXD register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
Id																															A	A	A	A	A	A	A	A				
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Id	RW	Field	Value Id	Value	Description																																					
A	R	RXD			RX data received in previous transfers, double buffered																																					

50.10.11 TXD

Address offset: 0x51C

## TXD register

## 50.10.12 BAUDRATE

Address offset: 0x524

## Baud rate

## **50.10.13 CONFIG**

Address offset: 0x56C

## Configuration of parity and hardware flow control

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Id	B	B	B	A																												
<b>Reset 0x00000000</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Id	RW	Field	Value Id	Value			Description																									
A	RW	HWFC					Hardware flow control																									
			Disabled	0			Disabled																									
			Enabled	1			Enabled																									
B	RW	PARITY					Parity																									
			Excluded	0x0			Exclude parity bit																									
			Included	0x7			Include parity bit																									

## **50.11 Electrical specification**

### 50.11.1 UART electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
$f_{UART}$	Baud rate for UART <sup>42</sup> .			1000	kbps
$I_{UART1M}$	Run current at max baud rate.		55		µA
$I_{UART115k}$	Run current at 115200 bps.		55		µA
$I_{UART1k2}$	Run current at 1200 bps.		55		µA
$I_{UART,IDLE}$	Idle current for UART		1		µA
$t_{UART,CTSH}$	CTS high time	1			µs
$t_{UART,START,LP}$	Time from STARTRX/STARTTX task to transmission started, low power mode		$t_{UART,START}$		µs
			+		
			$t_{START\_HFIN}$		
$t_{UART,START,CL}$	Time from STARTRX/STARTTX task to transmission started, constant latency mode	1			µs

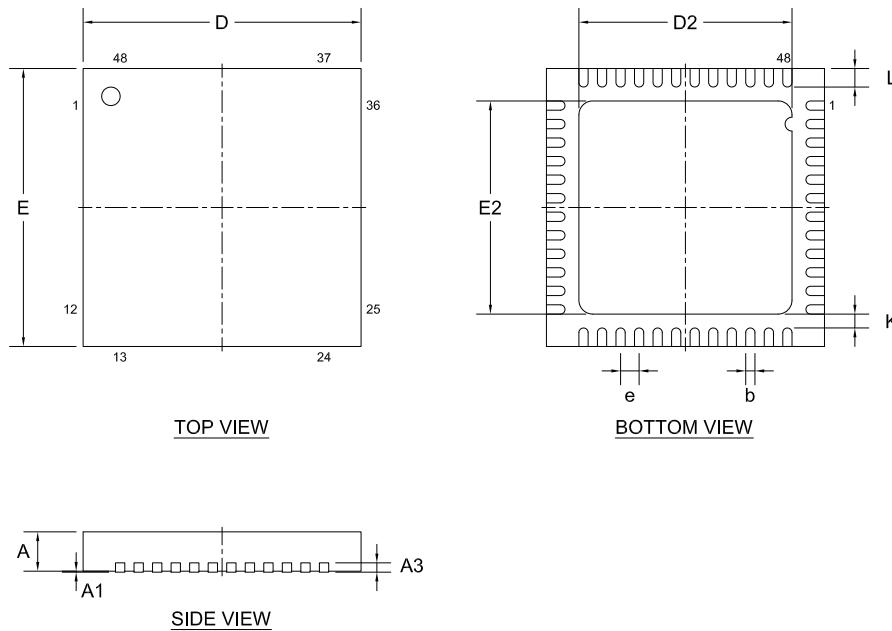
---

<sup>42</sup> Higher baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

# 51 Mechanical specifications

The mechanical specifications for the packages show the dimensions in millimeters.

## 51.1 QFN48 6 x 6 mm package

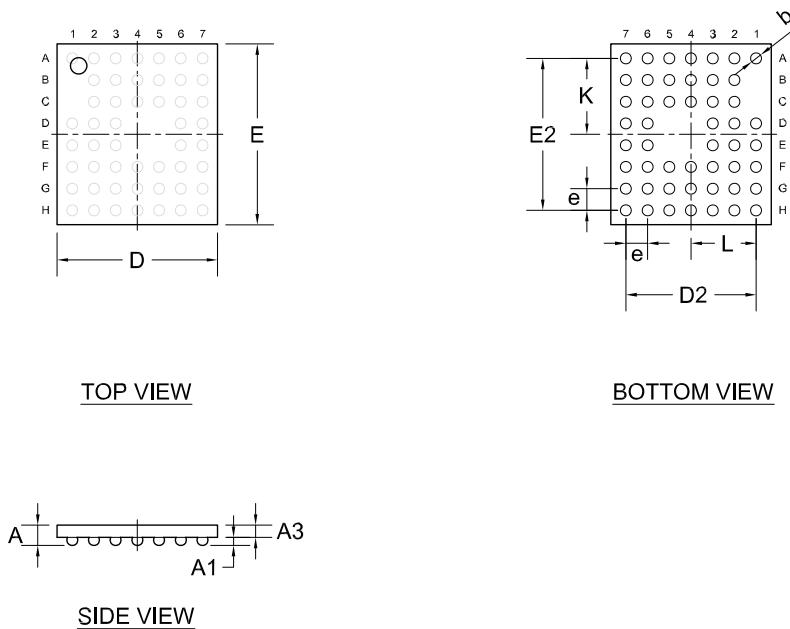


**Figure 163: QFN48 6 x 6 mm package**

**Table 130: QFN48 dimensions in millimeters**

Package	A	A1	A3	b	D, E	D2, E2	e	K	L	
	0.80	0.00		0.15		4.50		0.20	0.35	Min.
QFN48 (6x6)	0.85	0.02	0.2	0.20	6.0	4.60	0.4		0.40	Nom.
	0.90	0.05		0.25		4.70			0.45	Max.

## 51.2 WLCSP package



**Figure 164: WLCSP package**

**Table 131: WLCSP packet dimensions in millimeters**

Package	A	A1	A3	b	D	E	D2	E2	e	K	L	
	0.351	0.13		0.19								Min.
WLCSP (3.0 × 3.2)	0.375	0.15	0.225	0.20	2.956	3.226	2.4	2.8	0.4	1.4	1.2	Nom.
	0.399	0.17		0.25								Max.

## 52 Ordering information

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

### 52.1 IC marking

The nRF52832 IC package is marked like described below.

N	5	2	8	3	2
<P	P>	<V	V>	<H>	<P>
<Y	Y>	<W	W>	<L	L>

Figure 165: Package marking

### 52.2 Box labels

Here are the box labels used for the nRF52832.

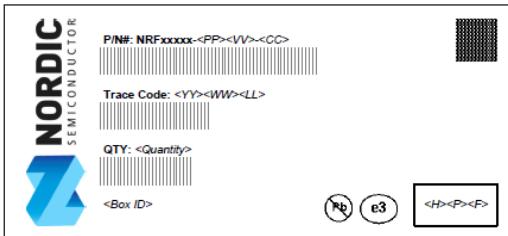


Figure 166: Inner box label

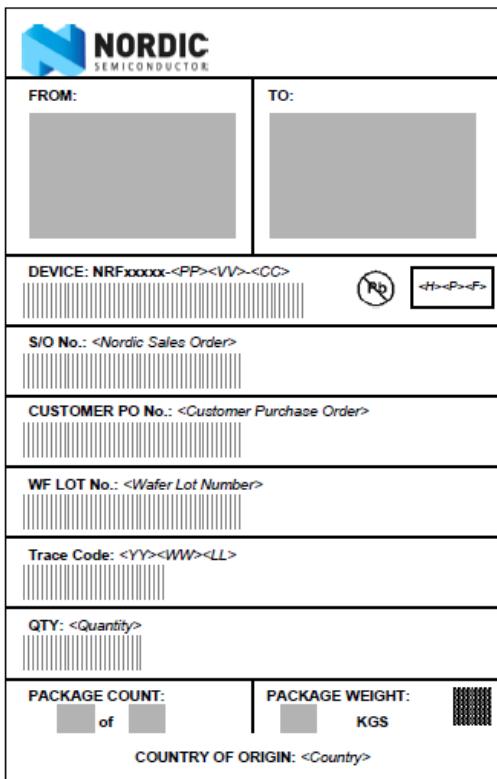


Figure 167: Outer box label

## 52.3 Order code

Here are the nRF52832 order codes and definitions.

n	R	F	5	2	8	3	2	-	<P	P>	<V	V>	-	<C	C>
---	---	---	---	---	---	---	---	---	----	----	----	----	---	----	----

Figure 168: Order code

Table 132: Abbreviations

Abbreviation	Definition and implemented codes
N52/nRF52	nRF52 Series product
832	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

## 52.4 Code ranges and values

Defined here are the nRF52832 code ranges and values.

**Table 133: Package variant codes**

<PP>	Package	Size (mm)	Pin/Ball count	Pitch (mm)
QF	QFN	6 x 6	48	0.4
CI	WLCSP	3.0 x 3.2	50	0.4

**Table 134: Function variant codes**

<VV>	Flash (kB)	RAM (kB)
AA	512	64
AB	256	32

**Table 135: Hardware version codes**

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

**Table 136: Production configuration codes**

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

**Table 137: Production version codes**

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

**Table 138: Year codes**

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

**Table 139: Week codes**

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

**Table 140: Lot codes**

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

**Table 141: Container codes**

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

## 52.5 Product options

Defined here are the nRF52832 product options.

**Table 142: nRF52832 order codes**

Order code	Minimum ordering quantity (MOQ)	Comment
nRF52832-QFAA-R7	1000	Availability to be announced.
nRF52832-QFAA-R	3000	
nRF52832-QFAA-T	490	
nRF52832-CIAA-R7	1500	
nRF52832-CIAA-R	7000	
nRF52832-QFAB-R	3000	
nRF52832-QFAB-R7	1000	
nRF52832-QFAB-T	490	

**Table 143: Development tools order code**

Order code	Description
nRF52-DK	nRF52 Development Kit

# 53 Reference circuitry

To ensure good RF performance when designing PCBs, it is highly recommended to use the PCB layouts and component values provided by Nordic Semiconductor.

Documentation for the different package reference circuits, including Altium Designer files, PCB layout files, and PCB production files can be downloaded from [Reference layout nRF52 Series](#).

## 53.1 Schematic QFAA and QFAB QFN48 with internal LDO setup

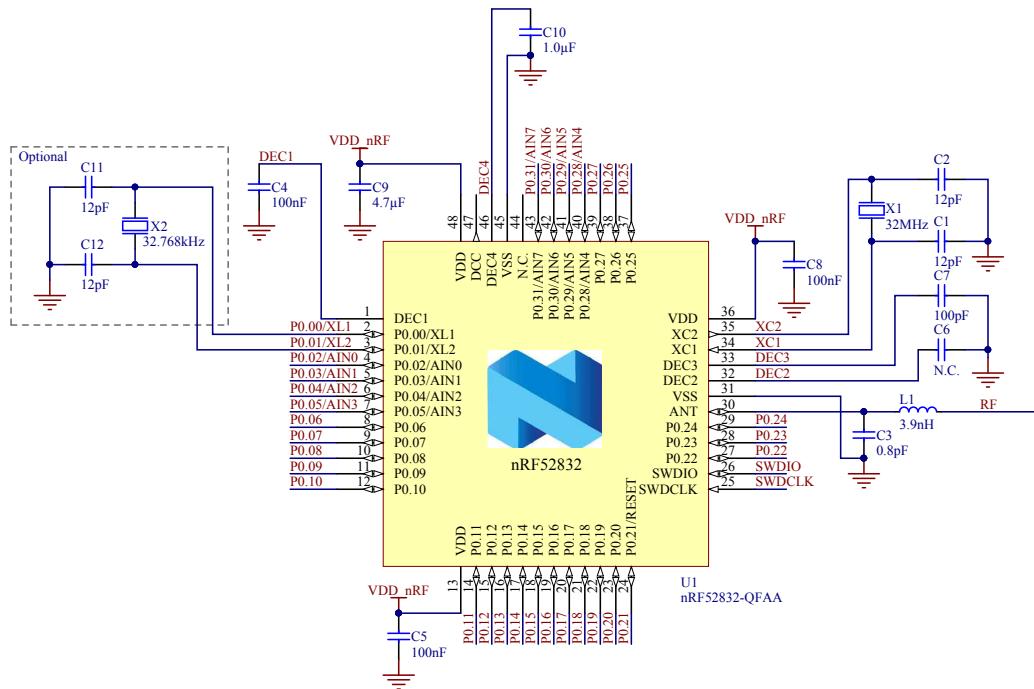


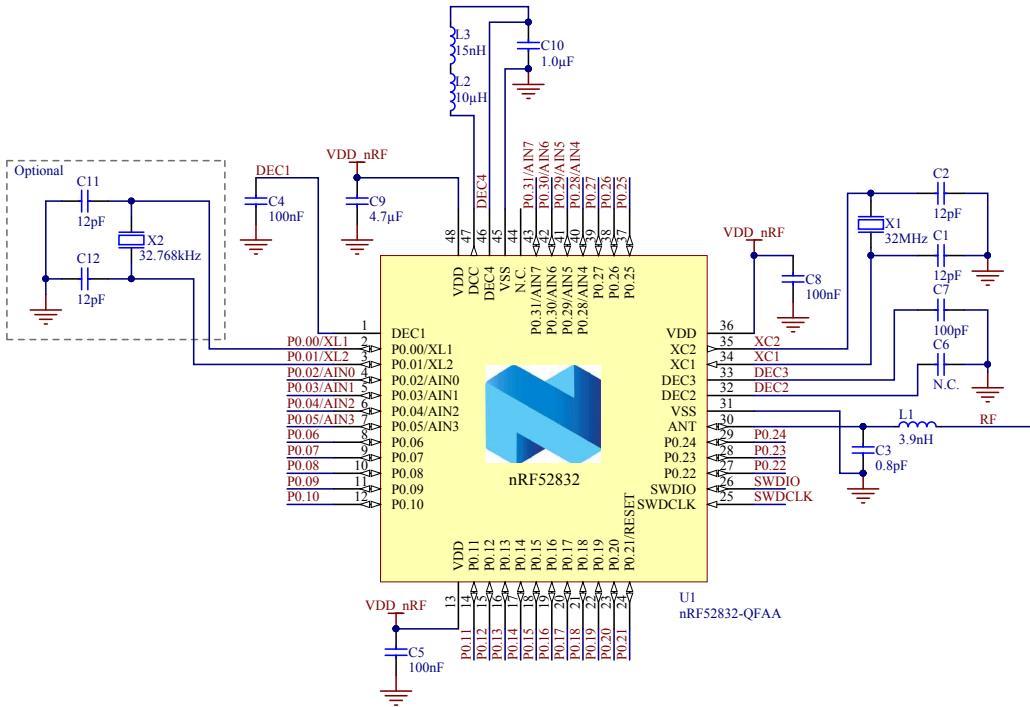
Figure 169: QFAA and QFAB QFN48 with internal LDO setup

For PCB reference layouts, see [Reference layout nRF52 Series](#).

Table 144: Bill of material for QFAA and QFAB QFN48 with internal LDO setup

Designator	Value	Description	Footprint
C1, C2, C11, C12	12 pF	Capacitor, NPO, ±2%	0402
C3	0.8 pF	Capacitor, NPO, ±5%	0402
C4, C5, C8	100 nF	Capacitor, X7R, ±10%	0402
C6	N.C.	Not mounted	0402
C7	100 pF	Capacitor, NPO, ±5%	0402
C9	4.7 μF	Capacitor, X5R, ±10%	0603
C10	1.0 μF	Capacitor, X7R, ±10%	0603
L1	3.9 nH	High frequency chip inductor ±5%	0402
U1	nRF52832-QFAA and nRF52832-QFAB	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	QFN-48
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, total tol. ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, total tol. ±50 ppm	XTAL_3215

## 53.2 Schematic QFAA and QFAB QFN48 with DC/DC regulator setup



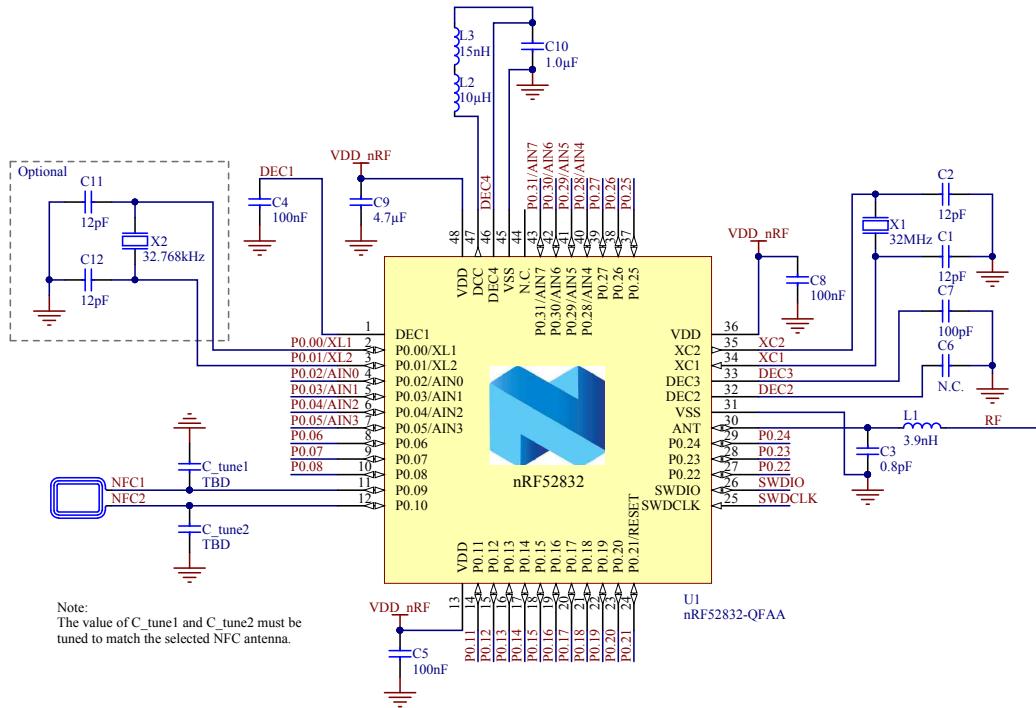
**Figure 170: QFAA and QFAB QFN48 with DC/DC regulator setup**

For PCB reference layouts, see [Reference layout nRF52 Series](#).

**Table 145: Bill of material for QFAA and QFAB QFN48 with DC/DC regulator setup**

Designator	Value	Description	Footprint
C1, C2, C11, C12	12 pF	Capacitor, NPO, ±2%	0402
C3	0.8 pF	Capacitor, NPO, ±5%	0402
C4, C5, C8	100 nF	Capacitor, X7R, ±10%	0402
C6	N.C.	Not mounted	0402
C7	100 pF	Capacitor, NPO, ±5%	0402
C9	4.7 µF	Capacitor, X5R, ±10%	0603
C10	1.0 µF	Capacitor, X7R, ±10%	0603
L1	3.9 nH	High frequency chip inductor ±5%	0402
L2	10 µH	Chip inductor, IDC,min = 50 mA, ±20%	0603
L3	15 nH	High frequency chip inductor ±10%	0402
U1	nRF52832-QFAA and nRF52832-QFAB	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	QFN-48
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, total tol. ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, total tol. ±50 ppm	XTAL_3215

### 53.3 Schematic QFAA and QFAB QFN48 with DC/DC regulator and NFC setup



**Figure 171: QFAA and QFAB QFN48 with DC/DC regulator and NFC setup**

For PCB reference layouts, see [Reference layout nRF52 Series](#).

**Table 146: Bill of material for QFAA and QFAB QFN48 with DC/DC converter and NFC setup**

Designator	Value	Description	Footprint
C1, C2, C11, C12	12 pF	Capacitor, NPO, $\pm 2\%$	0402
C3	0.8 pF	Capacitor, NPO, $\pm 5\%$	0402
C4, C5, C8	100 nF	Capacitor, X7R, $\pm 10\%$	0402
C6	N.C.	Not mounted	0402
C7	100 pF	Capacitor, NPO, $\pm 5\%$	0402
C9	4.7 $\mu$ F	Capacitor, X5R, $\pm 10\%$	0603
C10	1.0 $\mu$ F	Capacitor, X7R, $\pm 10\%$	0603
$C_{\text{tune}1}, C_{\text{tune}2}$	TBD pF	Capacitor, NPO, $\pm 5\%$	0402
L1	3.9 nH	High frequency chip inductor $\pm 5\%$	0402
L2	10 $\mu$ H	Chip inductor, IDC,min = 50 mA, $\pm 20\%$	0603
L3	15 nH	High frequency chip inductor $\pm 10\%$	0402
U1	nRF52832-QFAA and nRF52832-QFAB	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	QFN-48
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: $\pm 40$ ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 3215, 32.768 kHz, Cl=9 pF, $\pm 50$ ppm	XTAL_3215

### 53.4 Schematic CIAA WLCSP with internal LDO setup

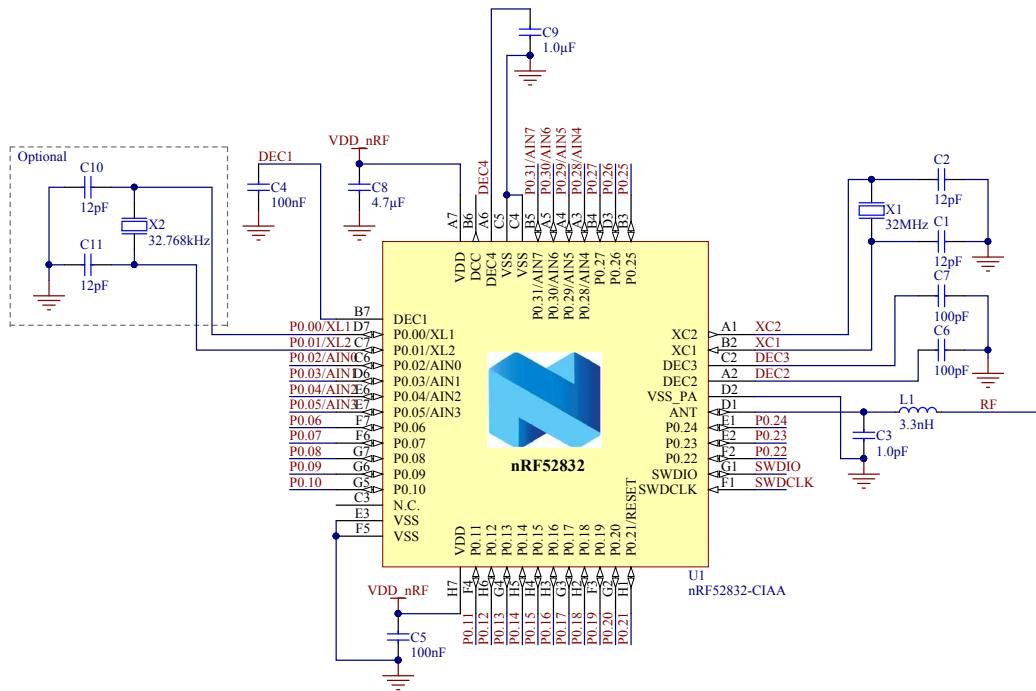


Figure 172: CIAA WLCSP with internal LDO setup

For PCB reference layouts, see [Reference layout nRF52 Series](#).

Table 147: Bill of material for CIAA WLCSP with internal LDO setup

Designator	Value	Description	Footprint
C1, C2, C10, C11	12 pF	Capacitor, NPO, ±2%	0201
C3	1.0 pF	Capacitor, NPO, ±5%	0201
C4, C5	100 nF	Capacitor, X7R, ±10%	0201
C6, C7	100 pF	Capacitor, NPO, ±5%	0201
C8	4.7 µF	Capacitor, X5R, ±10%	0603
C9	1.0 µF	Capacitor, X5R, ±5%	0402
L1	3.3 nH	High frequency chip inductor ±5%	0201
U1	nRF52832-CIAA	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	WLCSP_C50
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, total tol. ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 2012, 32.768 kHz, Cl=9 pF, ±50 ppm	XTAL_2012

### 53.5 Schematic CIAA WLCSP with DC/DC regulator setup

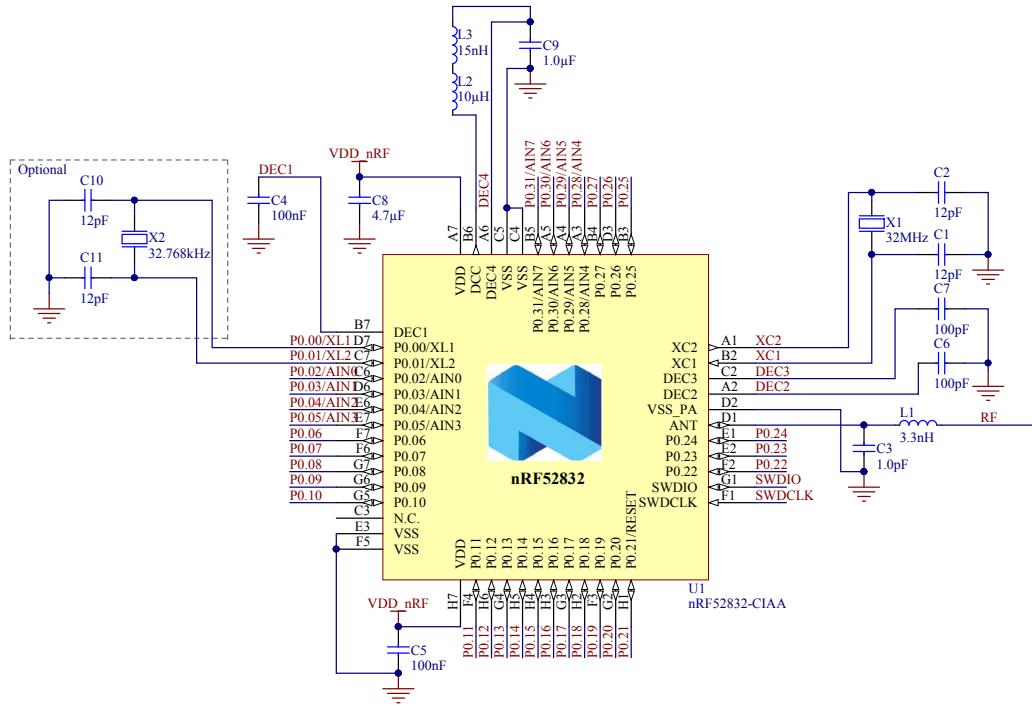


Figure 173: CIAA WLCSP with DC/DC regulator setup

For PCB reference layouts, see [Reference layout nRF52 Series](#).

Table 148: Bill of material for CIAA WLCSP with DC/DC regulator setup

Designator	Value	Description	Footprint
C1, C2, C10, C11	12 pF	Capacitor, NPO, ±2%	0201
C3	1.0 pF	Capacitor, NPO, ±5%	0201
C4, C5	100 nF	Capacitor, X7R, ±10%	0201
C6, C7	100 pF	Capacitor, NPO, ±5%	0201
C8	4.7 µF	Capacitor, X5R, ±10%	0603
C9	1.0 µF	Capacitor, X5R, ±5%	0402
L1	3.3 nH	High frequency chip inductor ±5%	0201
L2	10 µH	Chip inductor, IDC,min = 50 mA, ±20%	0603
L3	15 nH	High frequency chip inductor ±10%	0402
U1	nRF52832-CIAA	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	WLCSP_C50
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 2012, 32.768 kHz, Cl=9 pF, ±50 ppm	XTAL_2012

## 53.6 Schematic CIAA WLCSP with DC/DC regulator and NFC setup

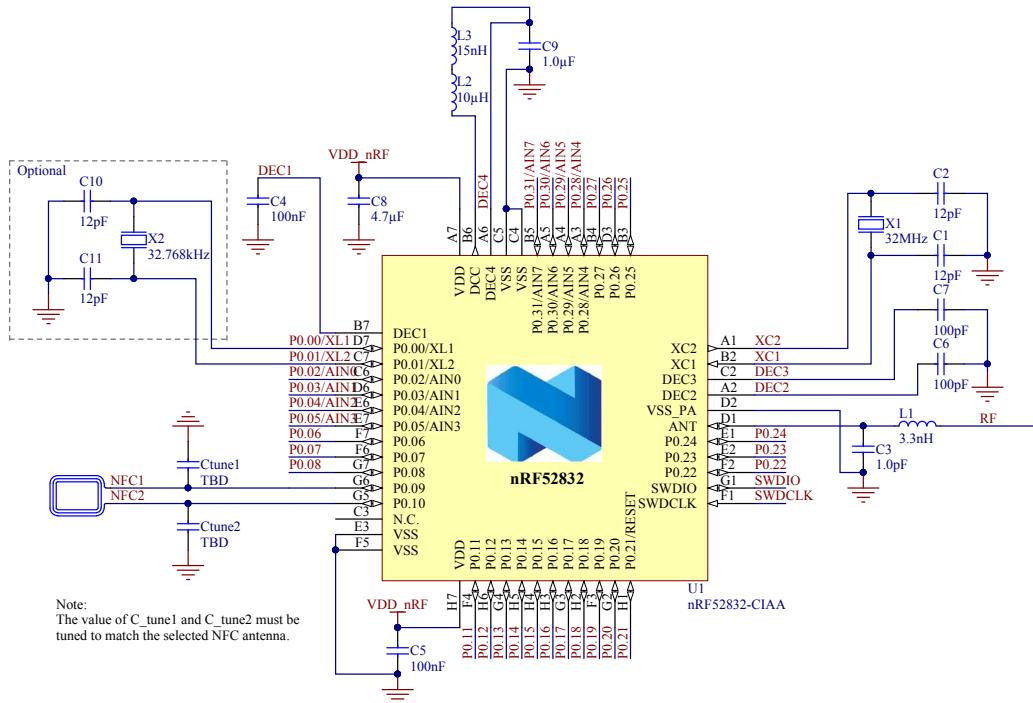


Figure 174: CIAA WLCSP with DC/DC regulator and NFC setup

For PCB reference layouts, see [Reference layout nRF52 Series](#).

Table 149: Bill of material for CIAA WLCSP with DC/DC converter and NFC setup

Designator	Value	Description	Footprint
C1, C2, C10, C11	12 pF	Capacitor, NPO, ±2%	0201
C3	1.0 pF	Capacitor, NPO, ±5%	0201
C4, C5	100 nF	Capacitor, X7R, ±10%	0201
C6, C7	100 pF	Capacitor, NPO, ±5%	0201
C8	4.7 µF	Capacitor, X5R, ±10%	0603
C9	1.0 µF	Capacitor, X5R, ±5%	0402
Ctune1, Ctune2	TBD pF	Capacitor, NPO, ±5%	0201
L1	3.3 nH	High frequency chip inductor ±5%	0201
L2	10 µH	Chip inductor, IDC,min = 50 mA, ±20%	0603
L3	15 nH	High frequency chip inductor ±10%	0402
U1	nRF52832-CIAA	Multi-protocol Bluetooth low energy, ANT, and 2.4 GHz proprietary system on chip	WLCSP_C50
X1	32 MHz	XTAL SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	XTAL SMD 2012, 32.768 kHz, Cl=9 pF, ±50 ppm	XTAL_2012

## 53.7 PCB guidelines

A well-designed PCB is necessary to achieve good RF performance. A poor layout can lead to loss in performance or functionality.

A qualified RF layout for the IC and its surrounding components, including matching networks, can be downloaded from [Reference layout nRF52 Series](#).

To ensure optimal performance, it is essential that you follow the schematics and layout references closely. Especially in the case of the antenna matching circuitry (components between device pin ANT and the antenna), any changes to the layout can change the behavior, resulting in degradation of RF performance or a need to change component values. All the reference circuits are designed for use with a 50 ohm single end antenna.

A PCB with a minimum of two layers, including a ground plane, is recommended for optimal performance. On PCBs with more than two layers, put a keep-out area on the inner layers directly below the antenna

matching circuitry (components between device pin ANT and the antenna) to reduce the stray capacitances that influence RF performance.

A matching network is needed between the RF pin ANT and the antenna, to match the antenna impedance (normally 50 ohm) to the optimum RF load impedance for the chip. For optimum performance, the impedance for the matching network should be set as described in the recommended package reference circuitry in [Reference circuitry](#) on page 545 above.

The DC supply voltage should be decoupled as close as possible to the VDD pins with high performance RF capacitors. See the schematics for recommended decoupling capacitor values. The supply voltage for the chip should be filtered and routed separately from the supply voltages of any digital circuitry.

Long power supply lines on the PCB should be avoided. All device grounds, VDD connections, and VDD bypass capacitors must be connected as close as possible to the IC. For a PCB with a topside RF ground plane, the VSS pins should be connected directly to the ground plane. For a PCB with a bottom ground plane, the best technique is to have via holes as close as possible to the VSS pads. A minimum of one via hole should be used for each VSS pin.

Fast switching digital signals should not be routed close to the crystal or the power supply lines. Capacitive loading of fast switching digital output lines should be minimized in order to avoid radio interference.

## 53.8 PCB layout example

The PCB layout shown below is a reference layout for the QFN package with internal LDO setup.

**Important:** Pay attention to how the capacitor C3 is grounded. It is not directly connected to the ground plane, but grounded via VSS pin 31. This is done to create additional filtering of harmonic components.

For all available reference layouts, see [Reference layout nRF52 Series](#).

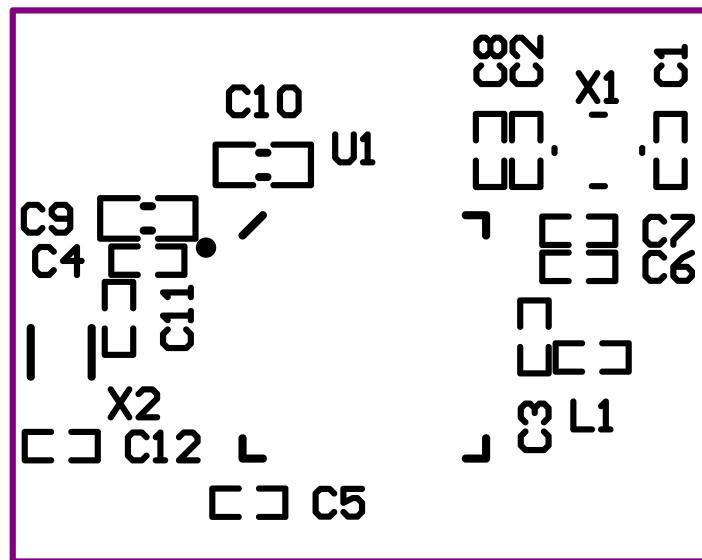


Figure 175: Top silk layer

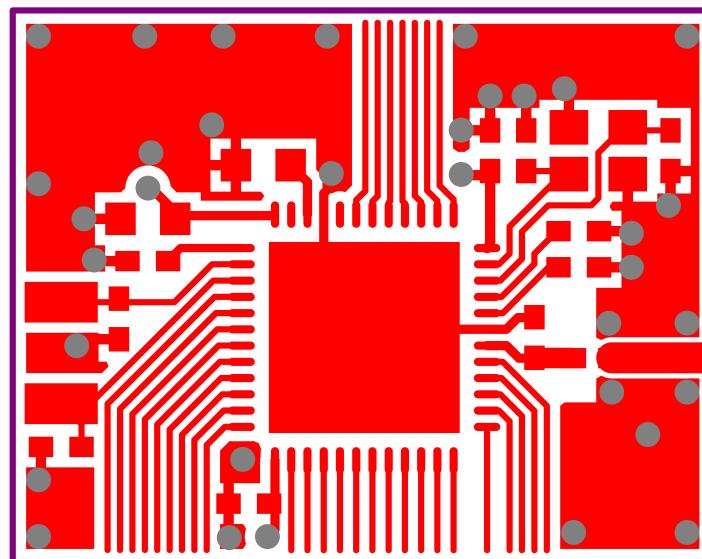


Figure 176: Top layer

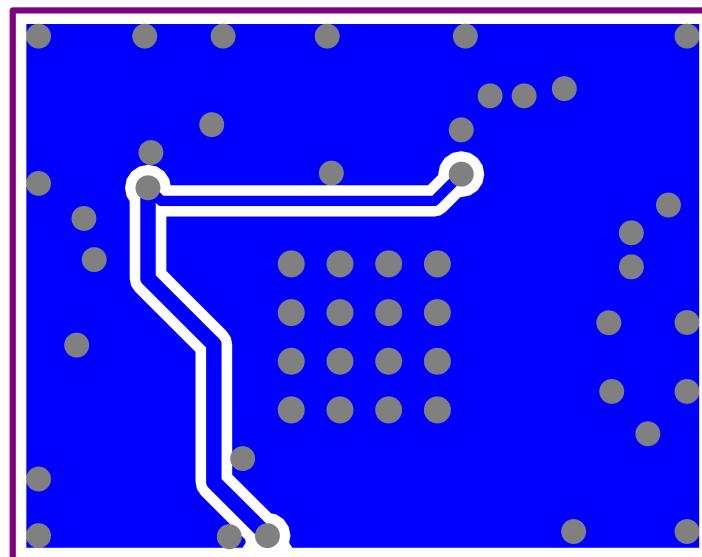


Figure 177: Bottom layer

**Important:** No components in bottom layer.

# 54 Liability disclaimer

## 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.

## 54.1 Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

### 54.1.1 RoHS and REACH statement

Nordic Semiconductor products meet the requirements of *Directive 2002/95/EC of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS)* 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](http://www.nordicsemi.com).

