nRF5340

Product Specification

v1.0

NORDIC® SEMICONDUCTOR

nRF5340 features

Features:

- 1.7 V to 5.5 V supply voltage range
- Single 32 MHz crystal operation
- Package variants
  - aQFN94™ package, 7 x 7 mm
- 1.8 V to 3.3 V regulated supply for external components
- Operating temperature from -40 to +105°C
- 48 general purpose I/O pins

Application core

- Arm® Cortex®-M33 with TrustZone® technology
  - 128 MHz or 64 MHz operation
  - 514 EEMBC CoreMark score running from flash memory, 4.0 CoreMark per MHz
  - Single-precision floating-point unit (FPU)
  - Digital signal processing (DSP) instructions
  - Data watchpoint and trace (DWT), embedded trace macrocell (ETM), instrumentation trace macrocell (ITM), and cross trigger interface (CTI)
  - Serial wire debug (SWD)
  - Trace port interface unit (TPIU)
    - 4-bit parallel trace of ITM and ETM trace data
    - Serial wire output (SWO) trace of ITM data
- 1 MB flash and 512 kB low leakage RAM
- Arm TrustZone CryptoCell™-312 security subsystem
  - NIST 800-90B, AIS-31, and FIPS 140-2 compliant random number generator
  - AES-128 and 256: ECB, CBC, CMAC/CBC-MAC, CTR, CCM/CCM*, GCM
  - SHA-1, SHA-2 up to 256 bits
  - Keyed-hash message authentication code (HMAC)
  - RSA public key cryptography with up to 3072-bit key size
  - ECC support for most used curves
  - Application key management using derived key model
- Two-way set associative cache towards flash and QSPI XIP code regions
- QSPI peripheral for communicating with an external flash memory device
  - Execute in place with optional on the fly encryption and decryption
- Near field communication (NFC-A) tag with wake-on field
  - Touch-to-pair support
- Up to 5x SPI master/slave with EasyDMA
- Up to 4x I^2C compatible two-wire master/slave with EasyDMA
- Up to 4x UART (CTS/RTS) with EasyDMA
  - Optional, built-in, flow control (CTS, RTS)
- Audio peripherals: I^2S, digital microphone interface (PDM)
- Up to 4x pulse width modulator (PWM) units with EasyDMA
- 12-bit, 200 ksps ADC with EasyDMA - eight configurable channels with programmable gain
- Up to 3x 32-bit timer with counter mode
- Up to 2x 24-bit real-time counter (RTC)
- Up to 2x Quadrature decoder (QDEC)

Network core

- Arm Cortex-M33
  - 64 MHz operation
  - 244 EEMBC CoreMark score running from flash memory, 101 CoreMark per mA
  - Cross trigger interface (CTI)
  - Serial wire debug (SWD)
  - SWO trace port
- 256 kB flash
- 64 kB low leakage RAM
- *Bluetooth*® 5.2, IEEE 802.15.4-2006, 2.4 GHz transceiver
  - -98 dBm sensitivity in 1 Mbps Bluetooth Low Energy mode
  - -104 dBm sensitivity in 125 kbps Bluetooth Low Energy mode (long range)
  - -101 dBm sensitivity in IEEE 802.15.4
  - -20 to +3 dBm configurable TX power
  - On-air compatible with nRF52, nRF51, nRF24L, and nRF24AP Series
  - Supported data rates:
    - *Bluetooth* 5.2 - 2 Mbps, 1 Mbps, 500 kbps, and 125 kbps
    - IEEE 802.15.4-2006 - 250 kbps
    - Proprietary 2.4 GHz - 2 Mbps, 1 Mbps
  - Angle of Arrival (AoA) and Angle of Departure (AoD) direction finding using *Bluetooth* Low Energy
  - Single-ended antenna output (on-chip balun)
  - 128-bit AES/ECB/CCM/AAR co-processor (on-the-fly packet encryption)
  - 3.2 mA run current in TX (0 dBm)
  - 2.6 mA run current in RX
  - RSSI (1 dB resolution)
- SPI master/slave with EasyDMA
- I^2C compatible two-wire master/slave with EasyDMA
- UART (CTS/RTS) with EasyDMA
- Up to 3x 32-bit timer with counter mode
- Up to 2x real-time counter (RTC)
- Temperature sensor
- Distributed programmable peripheral interconnect (DPPI)
- Inter-processor communication (IPC)
- Mutually exclusive peripheral (MUTEX)

Features:

- Distributed programmable peripheral interconnect (DPPI)
- Inter-processor communication (IPC)
- Mutually exclusive peripheral (MUTEX)

Applications:

- Advanced computer peripherals and I/O devices
  - Multi-touch trackpad
- Advanced wearables
  - Health/fitness sensor and monitor devices
  - Wireless payment enabled devices
- Wireless audio devices
  - Bluetooth Low Energy Audio
  - True wireless earbuds
  - Headphones, microphones, and speakers
- Internet of things (IoT)
  - Smart home sensors and controllers
  - Industrial IoT sensors and controllers
- Interactive entertainment devices
  - Remote controls
  - Gaming controllers
- Professional lighting
  - Wireless connected luminaire

Contents

nRF5340 features . . . ii

1 Revision history . . . 13

2 About this document . . . 14
2.1 Document status . . . 14
2.2 Peripheral chapters . . . 14
2.3 Register tables . . . 14
2.3.1 Fields and values . . . 15
2.3.2 Permissions . . . 15
2.4 Registers . . . 15
2.4.1 DUMMY . . . 15

3 Product overview . . . 17
3.1 Block diagram . . . 17
3.2 Memory . . . 18
3.2.1 RAM — Random access memory . . . 20
3.2.2 Flash — Non-volatile memory . . . 21
3.2.3 XIP — Execute in place . . . 21
3.2.4 Access latency . . . 21

4 Power and clock management . . . 22
4.1 System ON mode . . . 22
4.1.1 Power submodes . . . 23
4.2 System OFF mode . . . 23
4.2.1 Emulated System OFF mode . . . 23
4.3 Force-OFF mode . . . 24
4.3.1 Emulated Force-OFF mode . . . 24
4.4 Current consumption . . . 24
4.4.1 Electrical specification . . . 26
4.5 Power supply modes and regulators . . . 38
4.5.1 Normal voltage mode . . . 39
4.5.2 High voltage mode . . . 40
4.5.3 Power supply supervisor . . . 40
4.6 POWER — Power control . . . 43
4.6.1 Registers . . . 43
4.7 REGULATORS — Regulator control . . . 48
4.7.1 Normal voltage mode . . . 48
4.7.2 High voltage mode . . . 49
4.7.3 GPIO levels . . . 50
4.7.4 Registers . . . 51
4.7.5 Electrical specification . . . 53
4.8 USBREG — USB regulator control . . . 55
4.8.1 Registers . . . 56
4.8.2 Electrical specification . . . 60
4.9 VREQCTRL — Voltage request control . . . 60
4.9.1 Registers . . . 61
4.10 RESET — Reset control . . . 61
4.10.1 Power-on reset . . . 62

4.10.2 Pin reset . . . 62
4.10.3 Brownout reset . . . 63
4.10.4 Wakeup from System OFF mode reset . . . 63
4.10.5 Soft reset . . . 63
4.10.6 Watchdog timer reset . . . 63
4.10.7 Network Force-OFF . . . 63
4.10.8 Retained registers . . . 64
4.10.9 Application core reset behavior . . . 64
4.10.10 Network core reset behavior . . . 65
4.10.11 Registers . . . 66
4.10.12 Electrical specification . . . 68
4.11 CLOCK — Clock control . . . 69
4.11.1 HFCLK controller . . . 69
4.11.2 LFCLK controller . . . 73
4.11.3 Registers . . . 76
4.11.4 Electrical specification . . . 93
4.12 OSCILLATORS — Oscillator control . . . 94
4.12.1 High-frequency (32 MHz) crystal oscillator (HFXO) . . . 94
4.12.2 Low-frequency (32.768 kHz) crystal oscillator (LFXO) . . . 95
4.12.3 Low-frequency (32.768 kHz) external source . . . 96
4.12.4 Registers . . . 97
4.12.5 Electrical specification . . . 98

5 Application core . . . 100
5.1 Block diagram . . . 100
5.2 CPU . . . 102
5.2.1 Floating point interrupt . . . 102
5.2.2 Electrical specification . . . 102
5.2.3 CPU and support module configuration . . . 103
5.3 Memory . . . 103
5.3.1 Peripheral instantiation . . . 104
5.4 Core components . . . 108
5.4.1 CACHE — Instruction/data cache . . . 108
5.4.2 FICR — Factory information configuration registers . . . 117
5.4.3 UICR — User information configuration registers . . . 124
5.4.4 AHB multilayer . . . 129

6 Network core . . . 131
6.1 Block diagram . . . 131
6.2 CPU . . . 132
6.2.1 Electrical specification . . . 133
6.2.2 CPU and support module configuration . . . 133
6.3 Memory . . . 134
6.3.1 Peripheral instantiation . . . 135
6.4 Core components . . . 137
6.4.1 FICR — Factory information configuration registers . . . 137
6.4.2 UICR — User information configuration registers . . . 142
6.4.3 AHB multilayer . . . 143

7 Peripherals . . . 145
7.1 Peripheral interface . . . 145
7.1.1 Peripheral ID . . . 146
7.1.2 Peripherals with shared ID . . . 147

7.1.3 Peripheral registers . . . 147
7.1.4 Bit set and clear . . . 147
7.1.5 Tasks . . . 148
7.1.6 Events . . . 148
7.1.7 Publish and subscribe . . . 148
7.1.8 Shortcuts . . . 148
7.1.9 Interrupts . . . 148
7.1.10 Secure/non-secure peripherals . . . 149
7.2 EasyDMA . . . 149
7.2.1 EasyDMA error handling . . . 151
7.2.2 EasyDMA array list . . . 151
7.3 ACL — Access control lists . . . 152
7.3.1 Registers . . . 153
7.4 AAR — Accelerated address resolver . . . 154
7.4.1 Shared resources . . . 155
7.4.2 EasyDMA . . . 155
7.4.3 Resolving a resolvable address . . . 155
7.4.4 Example . . . 156
7.4.5 IRK data structure . . . 156
7.4.6 Registers . . . 156
7.4.7 Electrical specification . . . 162
7.5 CCM — AES CCM mode encryption . . . 162
7.5.1 Shared resources . . . 163
7.5.2 Keystream generation . . . 163
7.5.3 Encryption . . . 163
7.5.4 Decryption . . . 164
7.5.5 AES CCM and RADIO concurrent operation . . . 165
7.5.6 Encrypting packets on-the-fly in radio transmit mode . . . 165
7.5.7 Decrypting packets on-the-fly in RADIO receive mode . . . 166
7.5.8 CCM data structure . . . 167
7.5.9 EasyDMA and ERROR event . . . 168
7.5.10 Registers . . . 168
7.5.11 Electrical specification . . . 177
7.6 COMP — Comparator . . . 177
7.6.1 Shared resources . . . 179
7.6.2 Differential mode . . . 179
7.6.3 Single-ended mode . . . 180
7.6.4 Registers . . . 182
7.6.5 Electrical specification . . . 191
7.7 CRYPTOCELL — Arm TrustZone CryptoCell 312 . . . 192
7.7.1 Usage . . . 193
7.7.2 Direct memory access (DMA) . . . 194
7.7.3 Standards . . . 194
7.7.4 Registers . . . 195
7.8 DCNF — Domain configuration . . . 196
7.8.1 Protection . . . 196
7.8.2 Registers . . . 197
7.9 DPPI - Distributed programmable peripheral interconnect . . . 198
7.9.1 Subscribing to and publishing on channels . . . 199
7.9.2 DPPI configuration (DPPIC) . . . 201
7.9.3 Connection examples . . . 201
7.9.4 Special considerations for a system implementing TrustZone for Cortex-M processors . . . 202
7.9.5 Registers . . . 203
7.10 ECB — AES electronic codebook mode encryption . . . 206

7.10.1 Shared resources . . . 206
7.10.2 EasyDMA . . . 207
7.10.3 ECB data structure . . . 207
7.10.4 Registers . . . 207
7.10.5 Electrical specification . . . 211
7.11 EGU — Event generator unit . . . 211
7.11.1 Registers . . . 212
7.11.2 Electrical specification . . . 214
7.12 FPU - Floating point unit (FPU) exceptions . . . 214
7.12.1 Registers . . . 215
7.13 GPIO — General purpose input/output . . . 219
7.13.1 Assigning pins between cores, peripherals, or subsystems . . . 220
7.13.2 Pin configuration . . . 221
7.13.3 Pin sense mechanism . . . 222
7.13.4 GPIO security . . . 223
7.13.5 Registers . . . 225
7.13.6 Electrical specification . . . 229
7.14 GPIOTE — GPIO tasks and events . . . 230
7.14.1 Pin events and tasks . . . 231
7.14.2 Port event . . . 232
7.14.3 Tasks and events pin configuration . . . 232
7.14.4 Low power . . . 232
7.14.5 Registers . . . 233
7.14.6 Electrical specification . . . 238
7.15 I^2S — Inter-IC sound interface . . . 238
7.15.1 Mode . . . 239
7.15.2 Transmitting and receiving . . . 239
7.15.3 Left right clock (LRCK) . . . 240
7.15.4 Serial clock (SCK) . . . 241
7.15.5 Master clock (MCK) . . . 241
7.15.6 Width, alignment and format . . . 243
7.15.7 EasyDMA . . . 245
7.15.8 Module operation . . . 248
7.15.9 Pin configuration . . . 250
7.15.10 Registers . . . 250
7.15.11 Electrical specification . . . 264
7.16 IPC — Interprocessor communication . . . 264
7.16.1 IPC and PPI connections . . . 266
7.16.2 Registers . . . 267
7.16.3 Electrical specification . . . 270
7.17 KMU — Key management unit . . . 270
7.17.1 Functional view . . . 271
7.17.2 Access control . . . 271
7.17.3 Protecting the UICR content . . . 272
7.17.4 Usage . . . 272
7.17.5 Registers . . . 276
7.18 LPCOMP — Low-power comparator . . . 280
7.18.1 Shared resources . . . 281
7.18.2 Pin configuration . . . 282
7.18.3 Registers . . . 282
7.18.4 Electrical specification . . . 290
7.19 MUTEX — Mutual exclusive peripheral . . . 290
7.19.1 Registers . . . 292
7.20 NFCT — Near field communication tag . . . 292

7.20.1 Overview . . . 293
7.20.2 Operating states . . . 295
7.20.3 Pin configuration . . . 296
7.20.4 EasyDMA . . . 296
7.20.5 Frame assembler . . . 297
7.20.6 Frame disassembler . . . 298
7.20.7 Frame timing controller . . . 299
7.20.8 Collision resolution . . . 300
7.20.9 Antenna interface . . . 301
7.20.10 NFCT antenna recommendations . . . 301
7.20.11 Battery protection . . . 302
7.20.12 Digital Modulation Signal . . . 302
7.20.13 References . . . 303
7.20.14 Registers . . . 303
7.20.15 Electrical specification . . . 329
7.21 NVMC — Non-volatile memory controller . . . 329
7.21.1 Writing to flash . . . 329
7.21.2 Erasing a secure page in flash . . . 330
7.21.3 Erasing a non-secure page in flash . . . 330
7.21.4 Writing to user information configuration registers (UICR) . . . 330
7.21.5 Erase all . . . 330
7.21.6 NVMC protection mechanisms . . . 331
7.21.7 Cache . . . 332
7.21.8 Registers . . . 332
7.21.9 Electrical specification . . . 336
7.22 PDM — Pulse density modulation interface . . . 336
7.22.1 Master clock source selection . . . 337
7.22.2 Master clock generator . . . 337
7.22.3 Module operation . . . 338
7.22.4 Decimation filter . . . 338
7.22.5 EasyDMA . . . 339
7.22.6 Hardware example . . . 340
7.22.7 Pin configuration . . . 340
7.22.8 Registers . . . 341
7.22.9 Electrical specification . . . 350
7.23 PWM — Pulse width modulation . . . 350
7.23.1 Wave counter . . . 351
7.23.2 Decoder with EasyDMA . . . 354
7.23.3 Limitations . . . 362
7.23.4 Pin configuration . . . 362
7.23.5 Registers . . . 362
7.24 QDEC — Quadrature decoder . . . 373
7.24.1 Sampling and decoding . . . 374
7.24.2 LED output . . . 375
7.24.3 Debounce filters . . . 375
7.24.4 Accumulators . . . 376
7.24.5 Output/input pins . . . 376
7.24.6 Pin configuration . . . 376
7.24.7 Registers . . . 377
7.24.8 Electrical specification . . . 392
7.25 QSPI — Quad serial peripheral interface . . . 392
7.25.1 Configuring QSPI . . . 393
7.25.2 Write operation . . . 394
7.25.3 Read operation . . . 394

7.25.4 Erase operation . . . 394
7.25.5 Execute in place . . . 394
7.25.6 Encryption . . . 395
7.25.7 Sending custom instructions . . . 396
7.25.8 Deep power-down mode . . . 397
7.25.9 Instruction set . . . 397
7.25.10 Interface description . . . 398
7.25.11 Registers . . . 403
7.25.12 Electrical specification . . . 421
7.26 RADIO — 2.4 GHz radio . . . 422
7.26.1 Packet configuration . . . 423
7.26.2 Address configuration . . . 424
7.26.3 Data whitening . . . 424
7.26.4 CRC . . . 425
7.26.5 Radio states . . . 425
7.26.6 Transmit sequence . . . 426
7.26.7 Receive sequence . . . 428
7.26.8 Received signal strength indicator (RSSI) . . . 429
7.26.9 Interframe spacing (IFS) . . . 429
7.26.10 Device address match . . . 430
7.26.11 Bit counter . . . 430
7.26.12 Direction finding . . . 431
7.26.13 IEEE 802.15.4 operation . . . 436
7.26.14 EasyDMA . . . 445
7.26.15 Registers . . . 446
7.26.16 Electrical specification . . . 493
7.27 RNG — Random number generator . . . 499
7.27.1 Bias correction . . . 499
7.27.2 Speed . . . 499
7.27.3 Registers . . . 499
7.27.4 Electrical specification . . . 503
7.28 RTC — Real-time counter . . . 503
7.28.1 Clock source . . . 503
7.28.2 Resolution versus overflow and the prescaler . . . 503
7.28.3 Counter register . . . 504
7.28.4 Overflow . . . 505
7.28.5 Tick event . . . 505
7.28.6 Event control . . . 505
7.28.7 Capture . . . 506
7.28.8 Compare . . . 506
7.28.9 Task and event jitter/delay . . . 509
7.28.10 Registers . . . 511
7.28.11 Electrical specification . . . 519
7.29 SAADC — Successive approximation analog-to-digital converter . . . 519
7.29.1 Shared resources . . . 520
7.29.2 Overview . . . 520
7.29.3 Digital output . . . 521
7.29.4 Analog inputs and channels . . . 521
7.29.5 Operation modes . . . 522
7.29.6 EasyDMA . . . 523
7.29.7 Resistor ladder . . . 524
7.29.8 Reference . . . 525
7.29.9 Acquisition time . . . 525
7.29.10 Limits event monitoring . . . 526

7.29.11 Registers . . . 527
7.29.12 Electrical specification . . . 544
7.29.13 Performance factors . . . 545
7.30 SPIM — Serial peripheral interface master with EasyDMA . . . 545
7.30.1 SPI master transaction sequence . . . 546
7.30.2 D/CX functionality . . . 547
7.30.3 Pin configuration . . . 548
7.30.4 Shared resources . . . 549
7.30.5 EasyDMA . . . 549
7.30.6 Low power . . . 549
7.30.7 Registers . . . 550
7.30.8 Electrical specification . . . 564
7.31 SPIS — Serial peripheral interface slave with EasyDMA . . . 565
7.31.1 Shared resources . . . 566
7.31.2 EasyDMA . . . 566
7.31.3 SPI slave operation . . . 567
7.31.4 Semaphore operation . . . 568
7.31.5 Pin configuration . . . 569
7.31.6 Registers . . . 570
7.31.7 Electrical specification . . . 582
7.32 SPU — System protection unit . . . 584
7.32.1 General concepts . . . 584
7.32.2 Flash access control . . . 585
7.32.3 RAM access control . . . 588
7.32.4 Peripheral access control . . . 591
7.32.5 Pin access control . . . 593
7.32.6 DPPI access control . . . 593
7.32.7 External domain access control . . . 595
7.32.8 Arm TrustZone for Cortex-M ID allocation . . . 596
7.32.9 Registers . . . 597
7.33 SWI — Software interrupts . . . 608
7.33.1 Registers . . . 609
7.34 TEMP — Temperature sensor . . . 609
7.34.1 Registers . . . 609
7.34.2 Electrical specification . . . 616
7.35 TIMER — Timer/counter . . . 617
7.35.1 Capture . . . 618
7.35.2 Compare . . . 618
7.35.3 Task delays . . . 618
7.35.4 Task priority . . . 618
7.35.5 Registers . . . 619
7.35.6 Electrical specification . . . 626
7.36 TWIM — I^2C compatible two-wire interface master with EasyDMA . . . 626
7.36.1 Shared resources . . . 628
7.36.2 EasyDMA . . . 628
7.36.3 Master write sequence . . . 628
7.36.4 Master read sequence . . . 629
7.36.5 Master repeated start sequence . . . 630
7.36.6 Low power . . . 631
7.36.7 Master mode pin configuration . . . 631
7.36.8 Registers . . . 632
7.36.9 Electrical specification . . . 646
7.36.10 Pullup resistor . . . 647
7.37 TWIS — I^2C compatible two-wire interface slave with EasyDMA . . . 647

7.37.1 Shared resources . . . 650
7.37.2 EasyDMA . . . 650
7.37.3 TWIS responding to a read command . . . 650
7.37.4 TWIS responding to a write command . . . 651
7.37.5 Master repeated start sequence . . . 652
7.37.6 Terminating an ongoing TWI transaction . . . 653
7.37.7 Low power . . . 653
7.37.8 Slave mode pin configuration . . . 653
7.37.9 Registers . . . 654
7.37.10 Electrical specification . . . 667
7.38 UARTE — Universal asynchronous receiver/transmitter with EasyDMA . . . 668
7.38.1 EasyDMA . . . 669
7.38.2 Transmission . . . 669
7.38.3 Reception . . . 670
7.38.4 Error conditions . . . 671
7.38.5 Using the UARTE without flow control . . . 671
7.38.6 Parity and stop bit configuration . . . 671
7.38.7 Low power . . . 672
7.38.8 Pin configuration . . . 672
7.38.9 Registers . . . 672
7.38.10 Electrical specification . . . 690
7.39 USBD — Universal serial bus device . . . 690
7.39.1 USB device states . . . 691
7.39.2 USB terminology . . . 692
7.39.3 USB pins . . . 693
7.39.4 USBD power-up sequence . . . 693
7.39.5 USB pull-up . . . 694
7.39.6 USB reset . . . 694
7.39.7 USB suspend and resume . . . 695
7.39.8 EasyDMA . . . 696
7.39.9 Control transfers . . . 697
7.39.10 Bulk and interrupt transactions . . . 700
7.39.11 Isochronous transactions . . . 702
7.39.12 USB register access limitations . . . 705
7.39.13 Registers . . . 705
7.39.14 Electrical specification . . . 734
7.40 VMC — Volatile memory controller . . . 734
7.40.1 RAM power states . . . 734
7.40.2 Registers . . . 735
7.41 WDT — Watchdog timer . . . 737
7.41.1 Reload criteria . . . 737
7.41.2 Temporarily pausing the watchdog . . . 737
7.41.3 Watchdog reset . . . 737
7.41.4 Stopping the watchdog . . . 737
7.41.5 Registers . . . 738
7.41.6 Electrical specification . . . 744

8 Debug and trace . . . 745

8.1 DAP — Debug access port . . . 746
8.1.1 Registers . . . 746
8.1.2 Electrical specification . . . 747
8.2 Access port protection . . . 748
8.3 Debug Interface mode . . . 750
8.4 Real-time debug . . . 751

8.5 ROM tables . . . 751
8.6 Cross-trigger network . . . 754
8.7 Multidrop serial wire debug . . . 756
8.8 Trace . . . 756
8.9 Enabling the trace port . . . 757
8.10 CTRL-AP - Control access port . . . 757
8.10.1 Reset request . . . 758
8.10.2 Erase all . . . 759
8.10.3 Mailbox interface . . . 759
8.10.4 Disabling erase protection . . . 760
8.10.5 Disabling access port protection . . . 760
8.10.6 Debugger registers . . . 761
8.10.7 Registers . . . 765
8.11 CTI - Cross Trigger Interface . . . 769
8.11.1 Registers . . . 770
8.12 TAD - Trace and debug control . . . 779
8.12.1 Registers . . . 779

9 Hardware and layout . . . 783
9.1 Pin assignments . . . 783
9.1.1 aQFN94 pin assignments . . . 784
9.2 Mechanical specifications . . . 787
9.2.1 aQFN94 7 x 7 mm package . . . 787
9.3 Reference circuitry . . . 788
9.3.1 Circuit configuration no. 1 for QKAA aQFN94 . . . 789
9.3.2 PCB layout example . . . 790

10 Recommended operating conditions . . . 793

11 Absolute maximum ratings . . . 794

12 Ordering information . . . 795
12.1 Device marking . . . 795
12.2 Box labels . . . 795
12.3 Order code . . . 796
12.4 Code ranges and values . . . 797
12.5 Product options . . . 798

13 Legal notices . . . 800

1 Revision history

Date	Version	Description
December 2020	1.0	First release

2 About this document

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

2.1 Document status

The document status reflects the level of maturity of the document.

Document name	Description
Objective Product Specification (OPS)	Applies to document versions up to 1.0. This document contains target specifications for product development.
Product Specification (PS)	Applies to document versions 1.0 and higher. This document contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Table 1: Defined document names

2.2 Peripheral chapters

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.

The chapters describing peripherals may include the following information:

- A detailed functional description of the peripheral
- Register configuration for the peripheral
- Electrical specification tables, containing performance data which apply for the operating conditions described in Recommended operating conditions on page 793.

2.3 Register tables

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

2.3.1 Fields and values

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

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

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

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

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

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

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

2.3.2 Permissions

Different fields in a register might have different access permissions enforced by hardware.

The access permission for each register field is documented in the **Access** column in the following ways:

Access	Description	Hardware behavior
RO	Read-only	Field can only be read. A write will be ignored.
WO	Write-only	Field can only be written. A read will return an undefined value.
RW	Read-write	Field can be read and written multiple times.
W1	Write-once	Field can only be written once per reset. Any subsequent write will be ignored. A read will return an undefined value.
RW1	Read-write-once	Field can be read multiple times, but only written once per reset. Any subsequent write will be ignored.

Table 2: Register field permission schemes

2.4 Registers

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

Table 3: Register overview

2.4.1 DUMMY

Address offset: 0x514

Example of a register controlling a dummy feature

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

3 Product overview

nRF5340 is a wireless, ultra-low power multicore System on Chip (SoC), integrating two fully programmable Arm Cortex-M33 processors, advanced security features, a range of peripherals, and a multiprotocol 2.4 GHz transceiver. The transceiver supports Bluetooth Low Energy, ANT™, and 802.15.4 for, among others, Thread and Zigbee protocols. It also allows the implementation of proprietary 2.4 GHz protocols.

The two Arm Cortex-M33 processors share the power, clock, and peripheral architecture with Nordic Semiconductor nRF51, nRF52, and nRF91 Series of SoCs, ensuring minimal porting efforts. The application core is a full-featured Arm Cortex-M33 processor including DSP instructions and FPU and running at up to 128 MHz with 1 MB of flash and 512 kB of RAM. The option to run the application processor at 64 MHz allows the CPU to increase energy efficiency. The network core is an Arm Cortex-M33 processor with a reduced feature set, designed for ultra-low power operation. It runs at a fixed 64 MHz frequency and contains 256 kB of flash and 64 kB of RAM.

The peripheral set offers a variety of analog and digital functionality enabling single-chip implementation of a wide range of applications. Arm TrustZone technology, Arm CryptoCell-312, and supporting blocks for system protection and key management are embedded for the advanced security needed for IoT applications.

3.1 Block diagram

The block diagram illustrates the overall system. More detailed diagrams of the two cores, including pins and EasyDMA connectivity, can be found in Block diagram on page 100 and Block diagram on page 131.

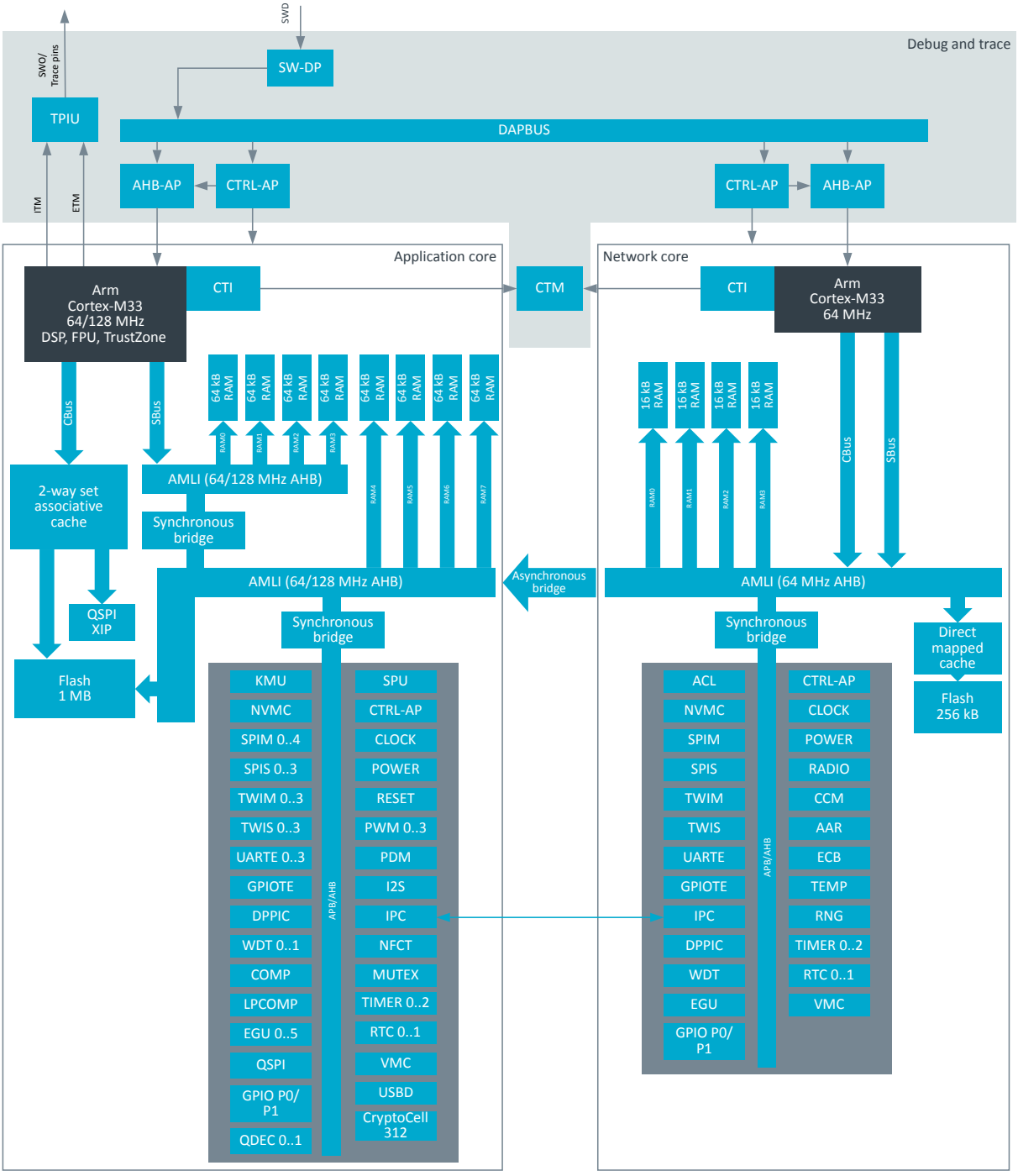

Figure 1: Simplified block diagram

3.2 Memory

The nRF5340 SoC contains two processor cores, each with flash and RAM that can be used for code and data storage.

Core	RAM	Flash
Application core	512 kB, arranged as follows: • 256 kB CPU single-cycle RAM • 256 kB of additional RAM	1024 kB in 4 kB pages
Network core	64 kB total	256 kB in 2 kB pages

Table 4: nRF5340 memory configuration

All memory and registers are found in the same address space, as shown in Memory map on page 20. This includes the two blocks of 256 kB RAM, which are accessible in the memory map as one contiguous 512 kB block of RAM. The first 256 kB block of RAM has single-cycle access time from the CPU, while up to four CPU cycles additional latency occurs when accessing the additional 256 kB block of RAM.

The application core memory is mapped to the network core memory map. This means that the network core CPU can access and use the application core memory for shared memory communication. The application core can restrict network core access through the domain configuration (DCNF) PROTECT registers, see DCNF — Domain configuration on page 196. Access to secure memory or peripherals as defined by the SPU — System protection unit on page 584 is also prevented when the network core is marked as non-secure in an application using TrustZone technology.

Note: The EasyDMA masters of the network core peripherals cannot access the application core RAM. The network core processor cannot execute code directly from the application core flash or QSPI XIP memories.

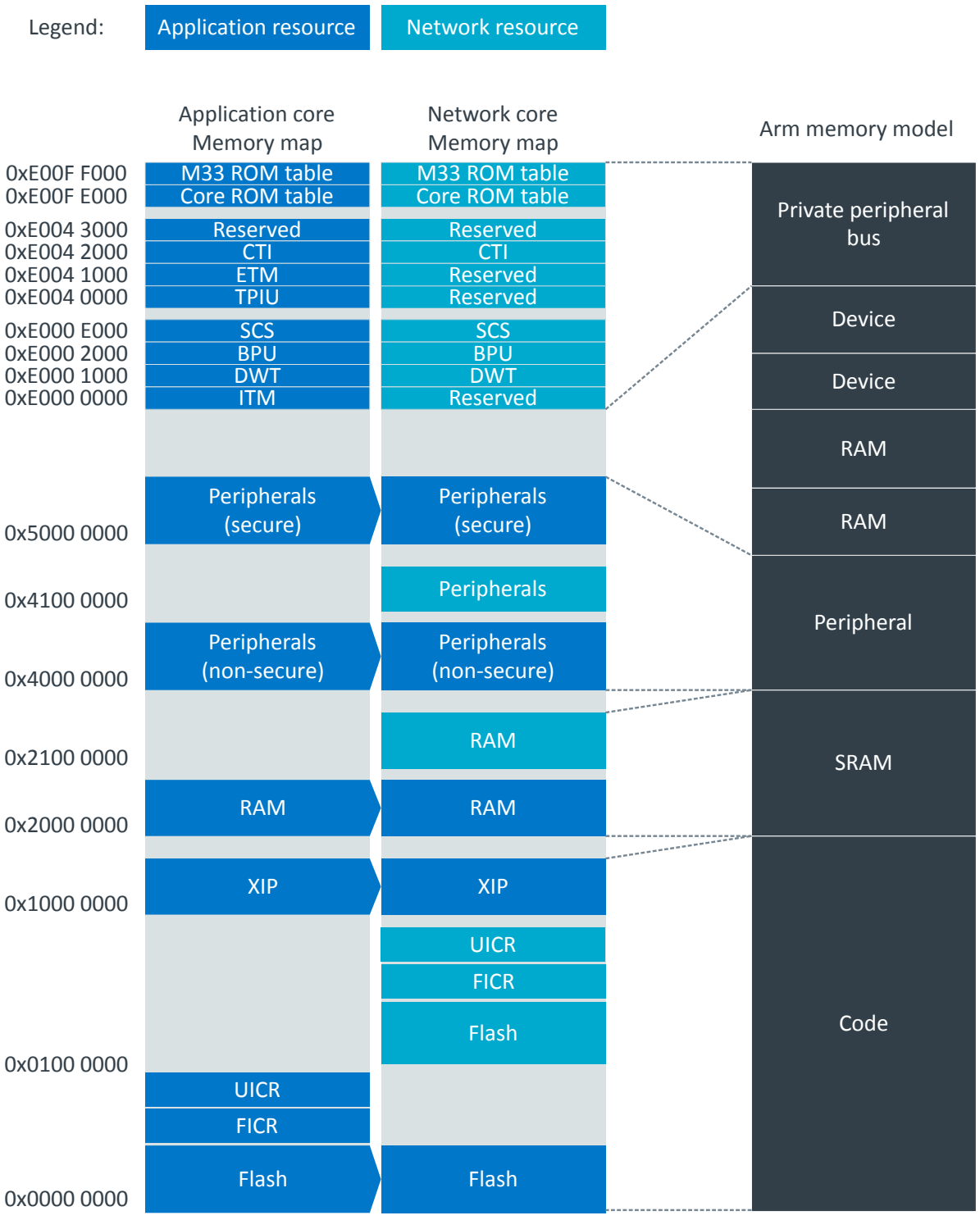

Figure 2: Memory map

3.2.1 RAM — Random access memory

RAM can be read and written an unlimited number of times.

Each RAM AHB slave within a core is connected to one or more RAM sections that has separate power control for System ON and System OFF mode operation. For details, see VMC — Volatile memory controller on page 734.

3.2.2 Flash — Non-volatile memory

Flash memory can be read an unlimited number of times by the CPU, but is restricted in the number of times it can be written to or erased, and in how it can be written.

Writing to flash memory is managed by the non-volatile memory controller (NVMC), see NVMC — Non-volatile memory controller on page 329.

Flash memory is divided in pages, as listed in nRF5340 memory configuration on page 19.

3.2.3 XIP — Execute in place

Execute in Place (XIP) allows the CPU to execute program code directly from the external flash memory device using the Quad serial peripheral interface (QSPI). The external flash memory contents can be on the fly encrypted and decrypted.

For details, see QSPI — Quad serial peripheral interface on page 392.

3.2.4 Access latency

When accessing memories or peripherals across bus bridges, additional access latency will occur. An example of such a case is network core access to application core memory or peripherals.

4 Power and clock management

The power and clock management system in nRF5340 is optimized for ultra-low power applications to ensure maximum power efficiency.

The core of the power and clock management system is the power management unit (PMU) shown in the following figure.

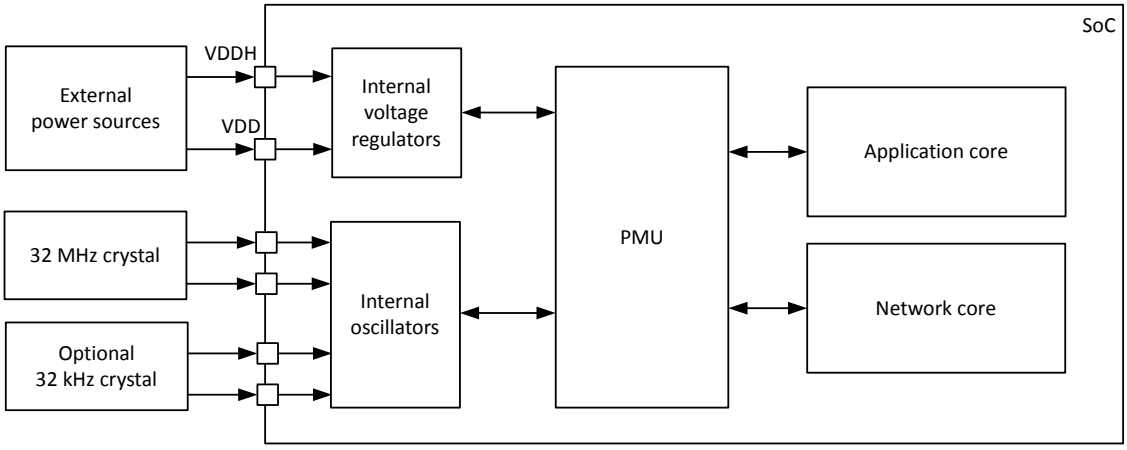

Figure 3: Power management unit

The PMU automatically tracks the power and clock resources required by the different components in the system at any given time. To achieve the lowest power consumption possible, the PMU optimizes the system by evaluating power and clock requests, automatically starting and stopping clock sources, and choosing regulator operation modes.

The nRF5340 start-up sequence after reset is described in RESET — Reset control on page 61.

4.1 System ON mode

System ON is the default operation mode after power-on reset.

In System ON, all functional blocks, such as the CPU and peripherals, can be in an idle or run state depending on the configuration set by the software and the state of the executing application. The network core's CPU and peripherals can be in an idle state, run state, or Force-OFF mode (see Force-OFF mode on page 24).

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

Voltage and frequency scaling

nRF5340 will automatically adjust the internal voltages to optimize power efficiency, which is a trade off between performance and power consumption.

Some configuration options will cause a higher internal voltage to be requested, which will be observed as an increase in power consumption. These configurations are the following:

- Setting the frequency of the application core's clock to 128 MHz, see Application core frequency scaling on page 72. The increased power consumption in this mode will also be observed when the CPU is sleeping, i.e. after executing the WFI (wait for interrupt) or WFE (wait for event) instructions.

Power consumption during System ON idle sleep is reduced by configuring the application core's clock to 64 MHz before entering CPU sleep.
- Using QSPI with 96 MHz clock frequency
- Using the USB peripheral
- When debugging
- Requesting additional voltage on the VREGRADIO supply using VREQCTRL — Voltage request control on page 60

4.1.1 Power submodes

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

The power submodes are:

- Constant latency
- Low-power

In Constant latency, the CPU wakeup latency and the PPI task response will be constant and kept at a minimum. This is secured by a set of resources that are always enabled. Compared to Low-power, the advantage of having a constant and predictable latency comes at a cost of increased power consumption. Constant latency is selected by triggering the CONSTLAT task.

In Low-power, the most power efficient supply option is chosen by the automatic power management system. Achieving the lowest power possible is at the expense of variations in CPU wakeup latency and PPI task response. Low-power is selected by triggering the LOWPWR task.

When the system enters System ON, it is by default in the Low-power submode.

4.2 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 register SYSTEMOFF on page 51 . The following signals/actions cause a wakeup from System OFF:

- The DETECT signal, generated by the GPIO peripheral
- The ANADETECT signal, generated by the LPCOMP peripheral
- The SENSE signal, generated by the NFCT peripheral to wake-on-field
- A valid USB voltage on the VBUS pin is detected
- A debug session is started
- A pin reset

When the device wakes up from System OFF, a system reset is performed. For more details, see Application core reset behavior on page 64.

One or more RAM sections can be retained in System OFF depending on the RAM retention settings in the peripheral VMC — Volatile memory controller on page 734.

Before entering System OFF, all on-going EasyDMA transactions should be completed. This is accomplished by making sure that the EasyDMA enabled peripheral is not active when entering System OFF. It is also recommended that the network core is in an idle state (i.e. peripherals are stopped and CPU is idle).

4.2.1 Emulated System OFF mode

When the device is in Debug Interface mode, System OFF is emulated to ensure that all required resources needed for debugging are available during System OFF.

Required resources needed for debugging include the following key components:

- Debug Interface mode on page 750
- CLOCK — Clock control on page 69
- POWER — Power control on page 43
- OSCILLATORS — Oscillator control on page 94
- REGULATORS — Regulator control on page 48
- RESET — Reset control on page 61
- NVMC — Non-volatile memory controller on page 329
- CPU
- Flash memory
- RAM

Because 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. For more information, see Debug and trace on page 745.

4.3 Force-OFF mode

Force-OFF is only applicable for the network core.

The register interface RESET — Reset control on page 61 is used by the application core to force the network core to Force-OFF mode. In this mode, the network core is stopped in order to achieve the lowest power consumption possible. When the network core is in Force-OFF mode, only the application core can release the mode, causing the network core to wake up and start the CPU again.

Before the application core sets the network core to Force-OFF mode, it is recommended that the network core is in an idle state:

- All peripherals are stopped.
- Any additional voltage on the VREGRADIO supply de-requested using VREQCTRL — Voltage request control on page 60.
- The CPU is an idle state, i.e. running the WFI or WFE instruction.

When the network core wakes up from Force-OFF mode, it is reset. For more details, see Network core reset behavior on page 65.

Several RAM sections can be retained in Force-OFF mode depending on the RAM retention settings in the peripheral VMC — Volatile memory controller on page 734.

4.3.1 Emulated Force-OFF mode

If the device is in Debug Interface mode, Force-OFF mode will be emulated to secure the required resources needed for debugging.

When Force-OFF mode is emulated, the CPU and all peripherals are reset. The CPU is prevented from running during debug access to a core's resources, including writing to RAM, flash, and/or peripherals. See Debug and trace on page 745 for more information.

4.4 Current consumption

Because the Power Management Unit (PMU) is constantly adjusting the different power and clock sources, estimating an application's current consumption can be challenging when the measurements cannot be performed directly on the hardware. To facilitate the estimation process, a set of current consumption scenarios is provided to show the typical current drawn from the VDD or VDDH supply.

Each scenario specifies a set of operations and conditions applying to the given scenario. All scenarios are listed in Electrical specification on page 26. The following table shows a set of common conditions used in all scenarios, unless otherwise stated in the description of a given scenario.

Condition	Value	Note
Supply	3 V on VDD/VDDH (normal voltage mode)	
Temperature	25ºC	
CPU	WFI (wait for interrupt)/WFE (wait for event) sleep	
Peripherals	All idle	
Clock	HFCLK = HFINT @ 64 MHz, LFCLK = Not running	
Regulator	DC/DC on VREGMAIN, VREGRADIO, and VREGH (when used)	
Application core RAM	8 kB	In System ON, RAM value refers to the amount of RAM that is switched on. The remainder of RAM is non retained. In System OFF, RAM value refers to amount of RAM that is retained.
Network core RAM	0 kB	
Cache enabled	Yes	Only applies when the CPU is running from flash memory.
Network core forced off	Yes	
32 MHz crystal	SMD 2016, 32 MHz, f_{tol} = ±30 ppm, C_L = 8 pF, R_S ≤ 50 Ω, D_L ≤ 100 µW	Only applies when the high frequency crystal oscillator (HFXO) is running. HFXO is used when the radio is running.
32 kHz crystal	SMD 2012, 32.768 kHz, f_{tol} = ±20 ppm, C_L = 9 pF, C_0 = 1.3 pF, R_L = 70 kΩ, D_L ≤ 1.0 µW	Only applies when the low frequency crystal oscillator (LFXO) is running.
Inductors	SMD 1608, L = 10 µH, Tol = 20%, I_{sat} ≥ 90 mA, R_{DC} ≤ 1.2 Ω	
Compiler version	GCC version 7.3.1 20180622 (arm-none-eabi-gcc)	
Compiler flags	`-mcpu=cortex-m33 -mthumb -mcmse -mfloat-abi=hard -mfpu=fpv5-sp-d16 -fno-delete-null-pointer-checks -fmax-errors=1 -funroll-all-loops -ffunction-sections -falign-functions=16 -fno-strict-aliasing -O3`	

Table 5: Current consumption scenarios, common conditions

4.4.1 Electrical specification

4.4.1.1 Sleep

Symbol	Description	Min.	Typ.	Max.	Units
I_{ON_IDLE1}	System ON, 0 kB application RAM, wake on any event		1.3		uA
$I_{ON\_IDLE1,LDO}$	System ON, 0k application RAM, wake on any event, regulator = LDO		3.3		uA
I_{ON_IDLE2}	System ON, wake on any event		1.3		uA
$I_{ON\_IDLE2,LDO}$	System ON, wake on any event, regulator = LDO		3.4		uA
I_{ON_IDLE3}	System ON, wake on any event, power-fail comparator enabled		1.3		uA
$I_{ON\_IDLE3,128MHz}$	System ON, wake on any event, power-fail comparator enabled, clock=HFINT128M		785		uA
I_{ON_IDLE4}	System ON, wake on GPIOTE input (event mode, LATENCY=LowLatency)		48		uA
$I_{ON\_IDLE4\_LP}$	System ON, wake on GPIOTE input (event mode, LATENCY=LowPower)		1.3		uA
I_{ON_IDLE5}	System ON, wake on GPIOTE PORT event		1.3		uA
I_{ON_IDLE6}	System ON, 0 kB application RAM, wake on RTC (running from LFXO clock)		1.5		uA
I_{ON_IDLE7}	System ON, wake on RTC (running from LFXO clock)		1.5		uA
I_{ON_IDLE8}	System ON, 0 kB application RAM, wake on RTC (running from LFXO clock), 5 V supply on VDDH, VREGH output = 3.3 V		1.7		uA
I_{ON_IDLE7}	System ON, 0 kB network RAM, wake on network RTC (running from LFXO clock)		1.5		uA
I_{ON_IDLE8}	System ON, 64 kB network RAM, wake on network RTC (running from LFXO clock)		1.7		uA
I_{ON_IDLE9}	System ON, 0 kB application RAM, wake on RTC (running from LFRC clock)		2.1		uA
I_{OFF0}	System OFF, 0 kB application RAM, wake on reset		1.0		uA
$I_{OFF0,LDO}$	System OFF, 0 kB application RAM, wake on reset; regulator = LDO		1.4		uA
I_{OFF1}	System OFF, 0 kB application RAM, wake on LPCOMP		0.9		uA

Symbol	Description	Min.	Typ.	Max.	Units
I_{OFF2}	System OFF, wake on reset		0.9		uA
I_{OFF3}	System OFF, 0 kB application RAM, wake on reset, 5 V supply on VDDH, VREGH output = 3.3V		1.1		uA
$I_{OFF3,LDO}$	System OFF, 0 kB application RAM, wake on reset, 5 V supply on VDDH, VREGH output = 3.3V; regulator = LDO		1.4		uA
I_{OFF4}	System OFF, 512 kB application RAM + 64 kB network RAM, wake on reset		2.4		uA

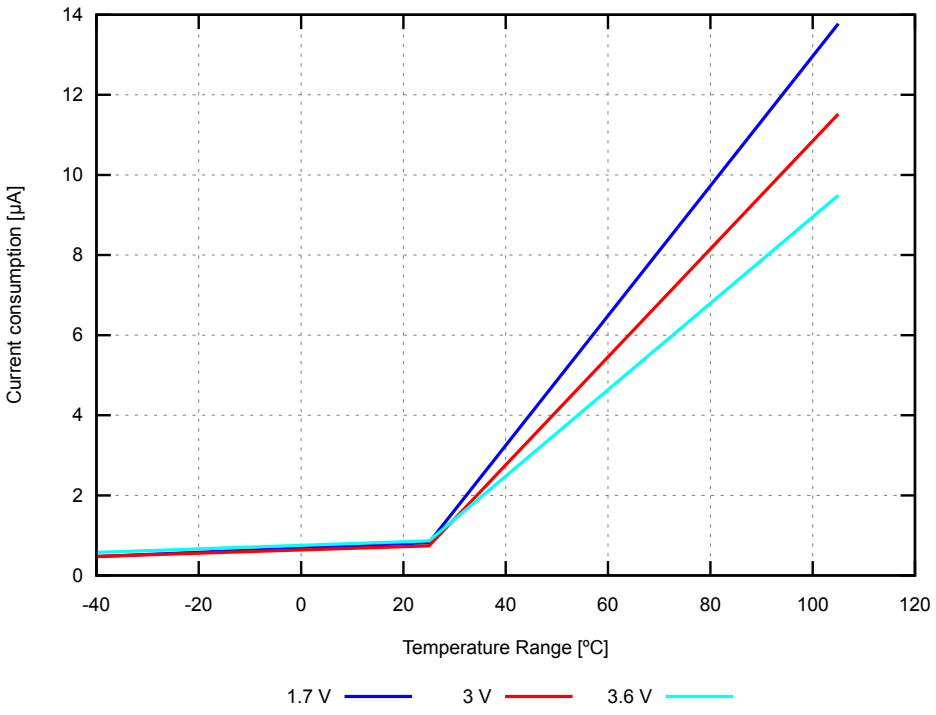

Figure 4: System OFF, 0 kB application RAM, wake on reset (typical values)

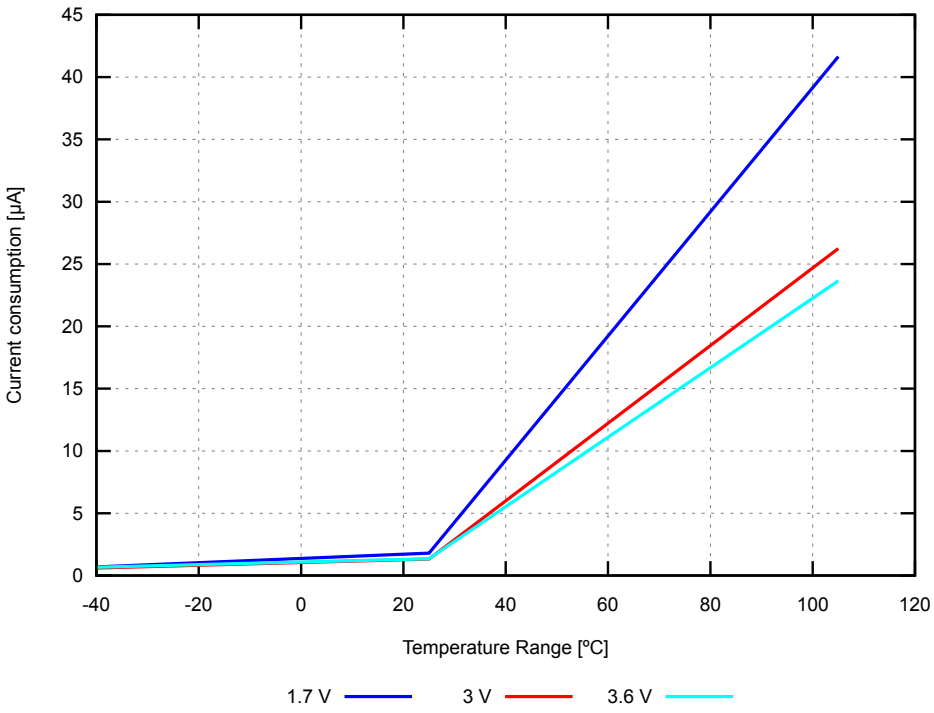

Figure 5: System ON, wake on any event, power-fail comparator enabled (typical values)

4.4.1.2 Application CPU running

The application CPU running parameters are obtained using the following compiler version:

Compiler: Arm version 6.14 (armclang)

Compiler flags:

```
-std=c99 --target=arm-arm-none-eabi -mcpu=cortex-m33 -mfpu=fpv5-sp-d16 -mfloat-abi=hard -
fno-rtti -flto -funsigned-char -mcmse -Omax -ffunction-sections
```

Linker flags:

```
-Omax
```

20 kB of RAM in application core switched on and retained in execute-from-flash cases, and 44 kB in execute-from-RAM cases.

Symbol	Description	Min.	Typ.	Max.	Units
$I_{APPCPU0}$	CPU running CoreMark from flash, regulator = LDO, clock = HFINT128M		15.5		mA
$I_{APPCPU2}$	CPU running CoreMark from flash, clock = HFXO128M		8.0		mA
$I_{APPCPU3}$	CPU running CoreMark from flash, clock = HFXO64M		3.6		mA
$I_{APPCPU4}$	CPU running CoreMark from flash, clock = HFINT128M		7.8		mA
$I_{APPCPU5}$	CPU running CoreMark from flash		3.3		mA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{APPCPU8}$	CPU running CoreMark from RAM, clock = HFINT128M		7.9		mA
$I_{APPCPU9}$	CPU running CoreMark from RAM		3.4		mA
$I_{APPCPU10}$	CPU running CoreMark from RAM, clock = HFXO128M		8.2		mA
$I_{APPCPU11}$	CPU running CoreMark from RAM, clock = HFXO64M		3.6		mA

4.4.1.3 Network CPU running

The network CPU running parameters are obtained using the following compiler version:

Compiler: Arm version 6.14 (armclang)

```
Compiler flags: -std=c99 --target=arm-arm-none-eabi -mcpu=cortex-m33+nodsp -mfpu=none -
mfloat-abi=soft -fno-rtti -flto -funsigned-char -Omax -ffunction-sections
```

Linker flags:

```
-Omax
```

20 kB of RAM in network core switched on and retained in execute-from-flash cases, and 40 kB in execute-from-RAM cases.

Clock and regulator settings only apply to network core. The settings in the application core are the same as the common conditions.

Symbol	Description	Min.	Typ.	Max.	Units
$I_{NETCPU0}$	CPU running CoreMark from flash, regulator = LDO		5.1		mA
$I_{NETCPU1}$	CPU running CoreMark from flash		2.4		mA
$I_{NETCPU2}$	CPU running CoreMark from flash, clock = HFXO64M		2.6		mA
$I_{NETCPU3}$	CPU running CoreMark from RAM, regulator = LDO		4.3		mA
$I_{NETCPU4}$	CPU running CoreMark from RAM		2.0		mA
$I_{NETCPU5}$	CPU running CoreMark from RAM, clock = HFXO64M		2.2		mA

4.4.1.4 COMP active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{COMP,LP}$	COMP enabled, Low-power mode		60		uA
$I_{COMP,NORM}$	COMP enabled, normal mode		62		uA
$I_{COMP,HS}$	COMP enabled, High-speed mode		68		uA

4.4.1.5 I2S active

Symbol	Description	Min.	Typ.	Max.	Units
I_{I2S0}	I2S transferring data @ 2 x 16 bit x 16 kHz (CONFIG.MCKFREQ = 32MDIV63, CONFIG.RATIO = 32X), clock = HFXO64M		2000		uA
I_{I2S1}	I2S transferring data @ 2 x 16 bit x 16 kHz (CONFIG.MCKFREQ = 510000, CONFIG.RATIO = 32X), clock = HFXO ACLK @ 12.288 MHz		2170		uA
I_{I2S2}	I2S transferring data @ 2 x 16 bit x 48 kHz (CONFIG.MCKFREQ = 505286656, CONFIG.RATIO = 32X), clock = HFXO ACLK @ 12.288 MHz		2310		uA

4.4.1.6 LPCOMP active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{LPCOMP,EN}$	LPCOMP enabled		45		uA

4.4.1.7 NFCT active

Symbol	Description	Min.	Typ.	Max.	Units
I_{SENSE}	System ON, current in SENSE STATE (this current does not apply when in NFC field)		1.3		uA
$I_{ACTIVATED}$	System ON, current in ACTIVATED STATE, clock = HFXO64M		1080		uA

4.4.1.8 PDM active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{PDM,RUN}$	PDM receiving and processing data @ 1 Msps (RATIO = 64, PDMCLKCTRL = 135274496), stereo mode, clock = HFXO64M		655		uA
$I_{PDM,RUN,ACLK}$	PDM receiving and processing data @ 1 Msps (RATIO = 64, PDMCLKCTRL = 343597056), stereo mode, HFXO ACLK = 12.288 MHz		1045		uA

4.4.1.9 PWM active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{PWM,RUN0}$	PWM running at 125 kHz, top = 10, duty = 50%		560		uA
$I_{PWM,RUN1}$	PWM running at 16 MHz, top = 10, duty = 50%		560		uA
$I_{PWM,RUN1,LDO}$	PWM running at 16 MHz, top = 10, duty = 50%; regulator = LDO		1035		uA
$I_{PWM,RUN2}$	PWM running at 125 kHz, top = 10, duty = 50%, clock = HFXO64M		750		uA
$I_{PWM,RUN3}$	PWM running at 16 MHz, top = 10, duty = 50%, clock = HFXO64M		755		uA

4.4.1.10 QDEC active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{QDEC,RUN}$	QDEC running		480		uA

4.4.1.11 QSPI active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{QSPI,IDLE}$	QSPI idle (enabled, but not activated)		45		uA
$I_{QSPI,ACTIVE}$	QSPI active (activated, but not transferring data)		1790		uA
$I_{QSPI,DATA}$	QSPI transferring data (activated, and transferring data to/ from external flash memory), SCKFREQ = 96 MHz, quad mode, clock = HFXO192M		4430		uA

4.4.1.12 RADIO transmitting/receiving

64 kB of network core RAM switched on and retained.

Clock and regulator settings only apply to network core. The settings in the application core are the same as the common conditions.

Symbol	Description	Min.	Typ.	Max.	Units
I_{RADIO_TX0}	Radio transmitting @ +3 dBm output power, 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		5.3		mA
I_{RADIO_TX1}	Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		4.1		mA
I_{RADIO_TX2}	Radio transmitting @ -40 dBm output power, 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		2.6		mA

Symbol	Description	Min.	Typ.	Max.	Units
I_{RADIO_TX3}	Radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M; regulator = LDO		9.2		mA
I_{RADIO_TX4}	Radio transmitting @ -40 dBm output power, 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M; regulator = LDO		5.0		mA
I_{RADIO_TX5}	Radio transmitting @ 0 dBm output power, 2 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		5.2		mA
I_{RADIO_TX6}	Radio transmitting @ 0 dBm output power, 500 kbps Bluetooth low energy (BLE) long-range (LR) mode, clock = HFXO64M		4.1		mA
I_{RADIO_TX7}	Radio transmitting @ 0 dBm output power, 125 kbps Bluetooth low energy (BLE) long-range (LR) mode, clock = HFXO64M		4.1		mA
I_{RADIO_TX8}	Radio transmitting @ 0 dBm output power, 250 kbps IEEE 802.15.4-2006 mode, clock = HFXO64M		4.1		mA
I_{RADIO_RX0}	Radio receiving @ 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		3.7		mA
I_{RADIO_RX1}	Radio receiving @ 1 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M; regulator = LDO		8.0		mA
I_{RADIO_RX2}	Radio receiving @ 2 Mbps Bluetooth low energy (BLE) mode, clock = HFXO64M		4.1		mA
I_{RADIO_RX3}	Radio receiving @ 500 kbps Bluetooth low energy (BLE) long-range (LR) mode, clock = HFXO64M		3.6		mA
I_{RADIO_RX4}	Radio receiving @ 125 kbps Bluetooth low energy (BLE) long-range (LR) mode, clock = HFXO64M		3.6		mA
I_{RADIO_RX5}	Radio receiving @ 250 kbps IEEE 802.15.4-2006 mode, clock = HFXO64M		3.9		mA

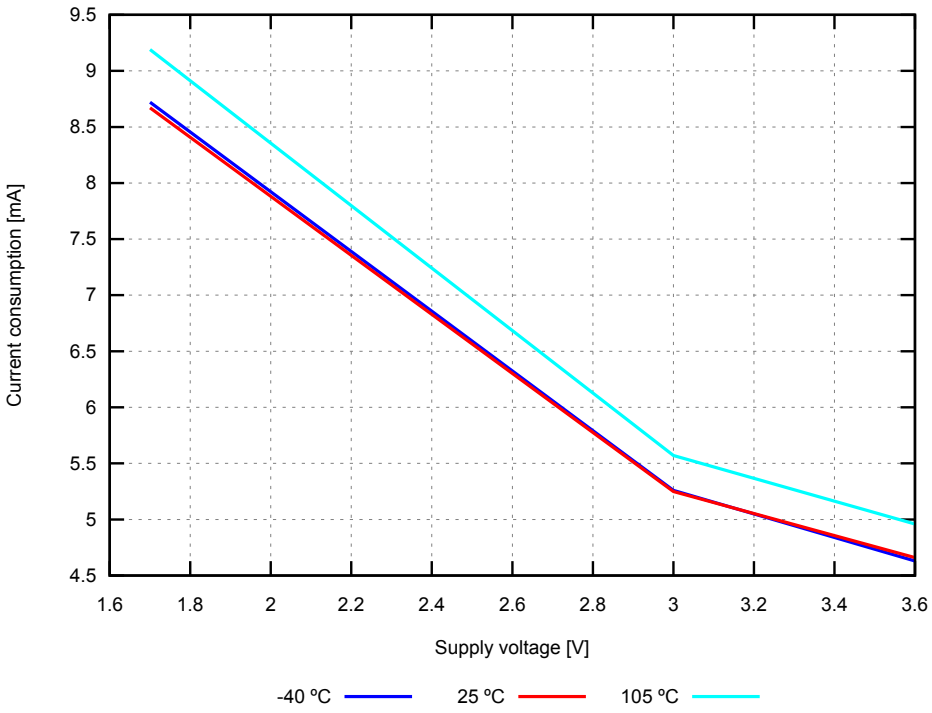

Figure 6: Radio transmitting at 3 dBm output power, 1 Mbps Bluetooth LE mode (typical values)

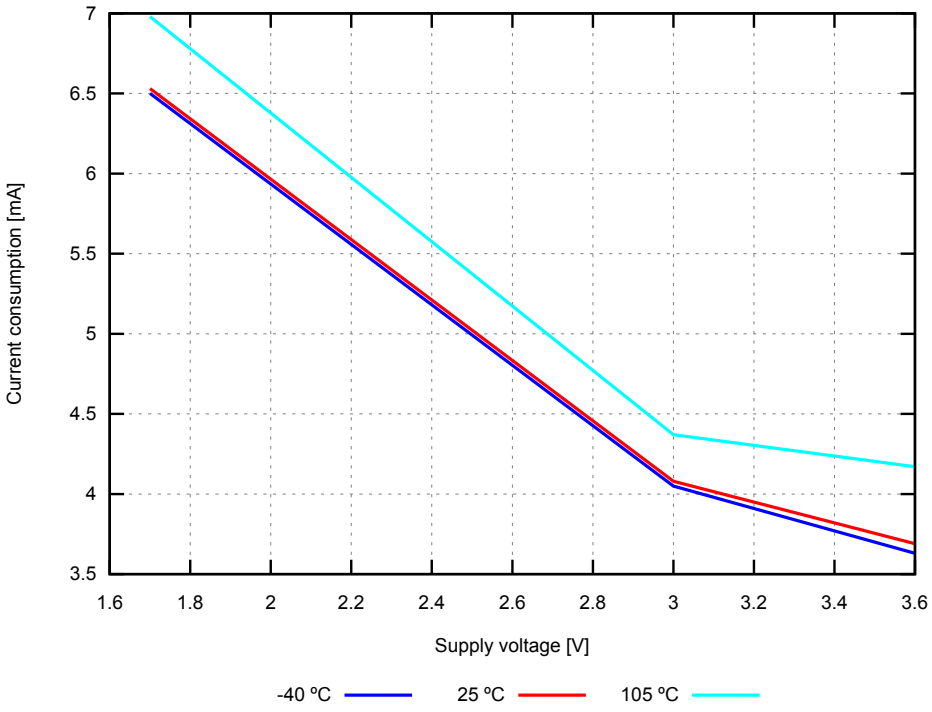

Figure 7: Radio transmitting at 0 dBm output power, 1 Mbps Bluetooth LE mode (typical values)

4.4.1.13 RNG active

Symbol	Description	Min.	Typ.	Max.	Units
I_{RNG0}	RNG running, 64 kB network RAM		270		uA

4.4.1.14 SAADC active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{SAADC,RUN}$	SAADC sampling @ 16 ksps, acquisition time = 20 us, clock = HFXO64M		980		uA
$I_{SAADC,TASK}$	SAADC sampling @ 1 kHz from RTC in task mode, LPOP=LowLat, acquisition time = 20 us, clock = HFINT64M and LFXO		770		uA
$I_{SAADC,TASK,LPOP}$	SAADC sampling @ 1 kHz from RTC in task mode,LPOP=LowPower , acquisition time = 20 us, clock = HFINT64M and LFXO		160		uA

4.4.1.15 TEMP active

Symbol	Description	Min.	Typ.	Max.	Units
I_{TEMP0}	TEMP started, 64 kB network RAM		615		uA

4.4.1.16 TIMER running

Symbol	Description	Min.	Typ.	Max.	Units
I_{TIMER0}	One TIMER running @ 1 MHz		475		uA
I_{TIMER1}	One TIMER running @ 1 MHz, clock = HFXO64M		670		uA
I_{TIMER2}	One TIMER running @ 16 MHz		560		uA
$I_{TIMER2,LDO}$	One TIMER running @ 16 MHz; regulator = LDO		1040		uA
I_{TIMER3}	One TIMER running @ 16 MHz, clock = HFXO64M		750		uA
$I_{TIMER3,LDO}$	One TIMER running @ 16 MHz, clock = HFXO64M; regulator LDO		1280		uA
I_{TIMER4}	One TIMER running @ 16 MHz, clock = HFINT128M		750		uA
$I_{NET,TIMER0}$	One network TIMER running @ 1 MHz		170		uA
$I_{NET,TIMER1}$	One network TIMER running @ 1 MHz, clock = HFXO64M		400		uA
$I_{NET,TIMER2}$	One network TIMER running @ 16 MHz		220		uA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{NET,TIMER3}$	One network TIMER running @ 16 MHz, clock = HFXO64M		445		uA

4.4.1.17 SPIM active

Symbol	Description	Min.	Typ.	Max.	Units
I_{SPIM0}	SPIM transferring data @ 2 Mbps		935		uA
I_{SPIM1}	SPIM transferring data @ 2 Mbps, clock = HFXO64M		1145		uA
I_{SPIM2}	SPIM transferring data @ 8 Mbps		1705		uA
I_{SPIM3}	SPIM transferring data @ 8 Mbps, clock = HFXO64M		1930		uA
I_{SPIM4}	SPIM transferring data @ 32 Mbps		2115		uA
I_{SPIM5}	SPIM transferring data @ 32 Mbps, clock = HFXO64M		2345		uA

4.4.1.18 SPIS active

Symbol	Description	Min.	Typ.	Max.	Units
I_{SPIS0}	SPIS configured and idle (enabled, no CSN activity)		145		uA
I_{SPIS1}	SPIS transferring data @ 2 Mbps		713		uA
I_{SPIS2}	SPIS transferring data @ 2 Mbps, clock = HFXO64M		913		uA

4.4.1.19 TWIM active

Symbol	Description	Min.	Typ.	Max.	Units
I_{TWIM0}	TWIM transferring data @ 100 kbps		965		uA
I_{TWIM1}	TWIM transferring data @ 100 kbps, clock = HFXO64M		1170		uA
I_{TWIM2}	TWIM transferring data @ 400 kbps		1000		uA
I_{TWIM3}	TWIM transferring data @ 400 kbps, clock = HFXO64M		1205		uA
I_{TWIM4}	TWIM transferring data @ 1000 kbps		2050		uA
I_{TWIM5}	TWIM transferring data @ 1000 kbps, clock = HFXO64M		2295		uA

4.4.1.20 TWIS active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{TWIS,IDLE}$	TWIS configured and enabled (IDLE state)		45		uA
I_{TWIS0}	TWIS transferring data @ 100 kbps		945		uA
I_{TWIS1}	TWIS transferring data @ 400 kbps		985		uA
I_{TWIS2}	TWIS transferring data @ 100 kbps, clock = HFXO64M		1150		uA
I_{TWIS3}	TWIS transferring data @ 400 kbps, clock = HFXO64M		1185		uA

4.4.1.21 UARTE active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{UARTE,IDLE0}$	UARTE RX idle (started, waiting for data, no data transfer)		645		uA
$I_{UARTE,IDLE1}$	UARTE RX idle (started, waiting for data, no data transfer), clock = HFXO64M		840		uA
I_{UARTE0}	UARTE transferring data @ 1200 bps, clock = HFXO64M		885		uA
I_{UARTE1}	UARTE transferring data @ 115200 bps, clock = HFXO64M		890		uA
I_{UARTE2}	UARTE receiving data @ 115200 bps, clock = HFXO64M		890		uA
I_{UARTE3}	UARTE transmitting and receiving data @ 115200 bps, clock = HFXO64M		895		uA

4.4.1.22 WDT active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{WDT,APP}$	Application MCU WDT started		2.0		uA
$I_{WDT,APP,LDO}$	Application MCU WDT started; regulator = LDO		4.9		uA
$I_{WDT,NET}$	Network MCU WDT started, 64 kB network RAM		3.2		uA

4.4.1.23 Compounded

These are scenarios where both cores are active. 20 kB of RAM in the application core and 64 kB of RAM in the network core are switched on and retained.

In scenarios where both cores are active, the clock and regulator settings apply to both.

Symbol	Description	Min.	Typ.	Max.	Units
I_{S0}	Application CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M		7.3		mA
I_{S1}	Application CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M		6.9		mA
I_{S2}	Application CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M, regulator = LDO		16.9		mA
I_{S3}	Application CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M; regulator = LDO		15.6		mA
I_{S4}	Application CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M, 5 V supply on VDDH, VREGH output = 3.3 V		6.7		mA
I_{S5}	Application CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M, 5 V supply on VDDH, VREGH output = 3.3 V		6.4		mA
I_{S6}	Network CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode, clock = HFXO64M		5.9		mA
I_{S7}	Network CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode, clock = HFXO64M		5.4		mA
I_{S8}	Application + Network CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M		9.1		mA
I_{S9}	Application + Network CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M		8.6		mA
I_{S10}	Application + Network CPU running CoreMark from flash, radio transmitting @ +3 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M		9.1		mA
I_{S11}	Application + Network CPU running CoreMark from flash, radio transmitting @ +3 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M; regulator = LDO		21.5		mA
I_{S12}	Application + Network CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M; regulator = LDO		20.2		mA

Symbol	Description	Min.	Typ.	Max.	Units
I_{S13}	Application + Network CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode; clock = HFXO64M; regulator = LDO		21.5		mA
I_{S14}	Network CPU running CoreMark from flash, radio transmitting @ 0 dBm output power, 1 Mbps Bluetooth Low Energy mode, clock = HFXO64M; regulator = LDO		12.6		mA
I_{S15}	Network CPU running CoreMark from flash, radio receiving @ 1 Mbps Bluetooth Low Energy mode, clock = HFXO64M; regulator = LDO		14.0		mA

4.4.1.24 USBD active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,ACTIVE,VBUS}$	Current from VBUS supply, USB active		1.2		mA
$I_{USB,SUSPEND,VBUS}$	Current from VBUS supply, USB suspended, CPU sleeping		180		uA
$I_{USB,ACTIVE,VDD}$	Current from VDD supply (normal voltage mode), all RAM retained, CPU running, USB active		3.0		mA
$I_{USB,SUSPEND,VDD}$	Current from VDD supply (normal voltage mode), all RAM retained, CPU sleeping, USB suspended		815		uA
$I_{USB,SUSPEND,VDD,LDO}$	Current from VDD supply (normal voltage mode), all RAM retained, CPU sleeping, USB suspended, regulator = LDO		135		uA
$I_{USB,ACTIVE,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (VREGH output), all RAM retained, CPU running, USB active		3.2		mA
$I_{USB,SUSPEND,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (VREGH output), all RAM retained, CPU sleeping, USB suspended		2340		uA
$I_{USB,SUSPEND,VDDH,LDO}$	Current from VDDH supply (high voltage mode), VDD=3 V (VREGH output), all RAM retained, CPU sleeping, USB suspended, regulator = LDO		125		uA
$I_{USB,DISABLED,VDD}$	Current from VDD supply, USB disabled, VBUS supply connected, all RAM retained, CPU sleeping		3		uA

4.5 Power supply modes and regulators

nRF5340 supports two different power supply voltage ranges, each having a corresponding power supply pin. The PMU automatically activates the correct voltage regulator depending on which power supply pin is used.

The nRF5340 PMU controls three different regulators to support the following power supply modes:

- Normal voltage mode on page 39 - Powers the device through the VDD pin

- High voltage mode on page 40 - Powers the device through the VDDH pin

In addition, the nRF5340 has a dedicated regulator used only for USB, controlled and operated separately using USBREG — USB regulator control on page 55.

These regulators and how they are connected to the supply pins are shown in the following figure.

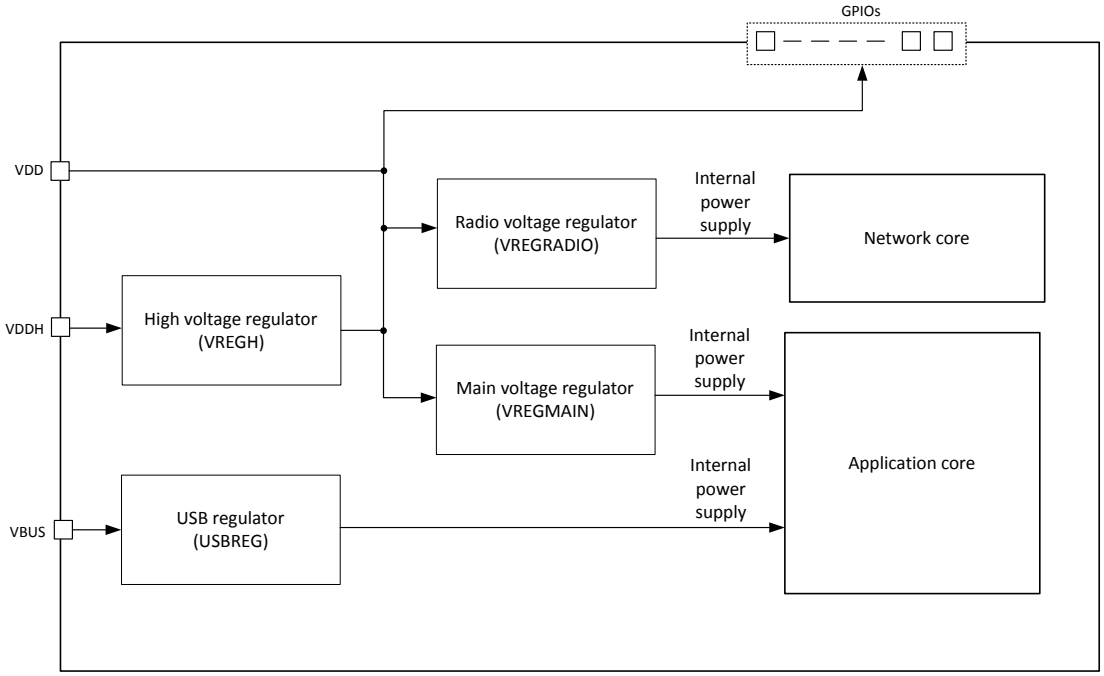

Figure 8: Regulators used in nRF5340

4.5.1 Normal voltage mode

When the device operates in normal voltage mode, only the main voltage regulator (VREGMAIN) and the radio voltage regulator (VREGRADIO) are used.

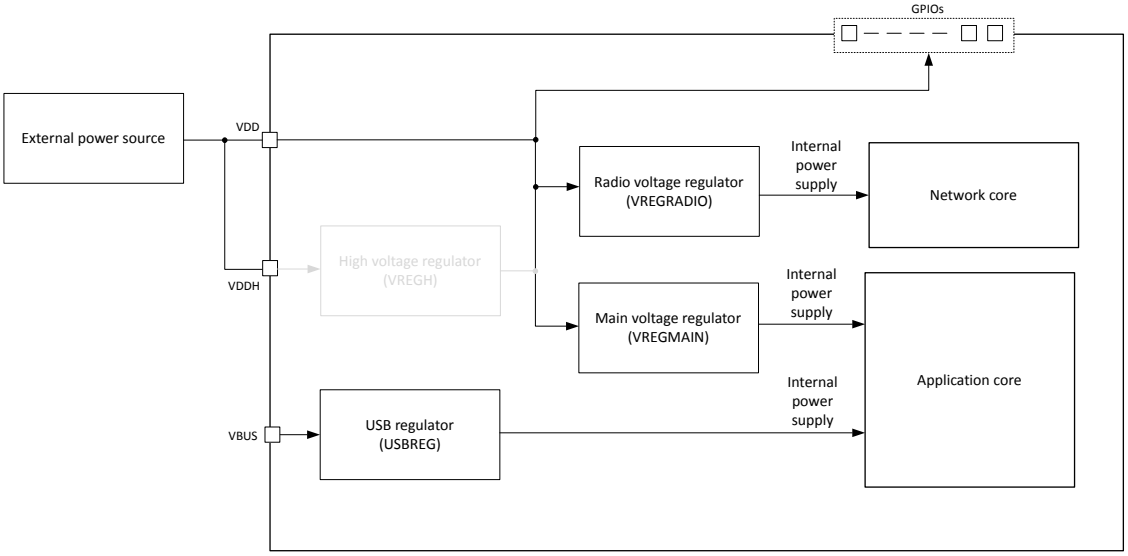

Figure 9: Regulator usage in normal voltage mode

The VDD and VDDH pins are connected together. The external power supply is connected to the both pins. In this case, the VREGH regulator is automatically deactivated.

In normal voltage mode, each regulator can operate in LDO or DC/DC mode. See Normal voltage mode on page 48 for details about configuration of the regulators in this mode.

4.5.2 High voltage mode

When the device operates in high voltage mode, the high voltage regulator (VREGH), the main voltage regulator (VREGMAIN) and the radio voltage regulator (VREGRADIO) are used.

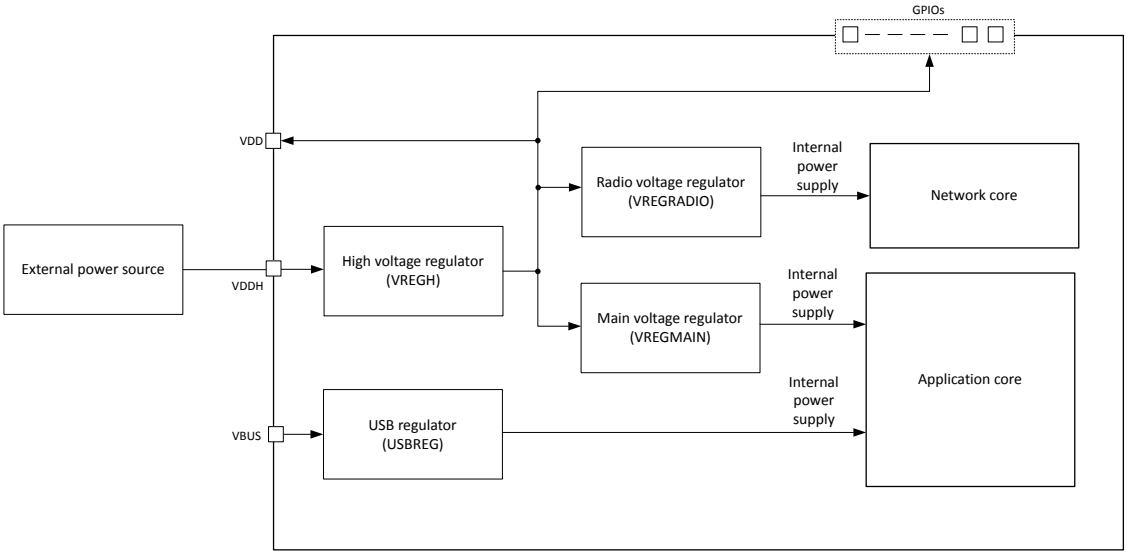

Figure 10: Regulator usage in high voltage mode

The external power supply is connected to the VDDH pin. The VREGMAIN and VREGRADIO regulators are used to power the internal circuitry from the VDD pin. The VDD pin is supplied by the VREGH regulator.

By default, the high voltage regulator is configured to also source external components from the VDD pin. To save power this feature should be disabled. For details, see High voltage mode on page 49.

In high voltage mode, each of the three regulators can operate in LDO or DC/DC mode. See High voltage mode on page 49 for details about configuring the regulators in this mode.

4.5.3 Power supply supervisor

Several voltage monitoring devices, enabled through the power management unit (PMU), are used to monitor the connected power supply.

The following figure illustrates the main components for power supply supervision.

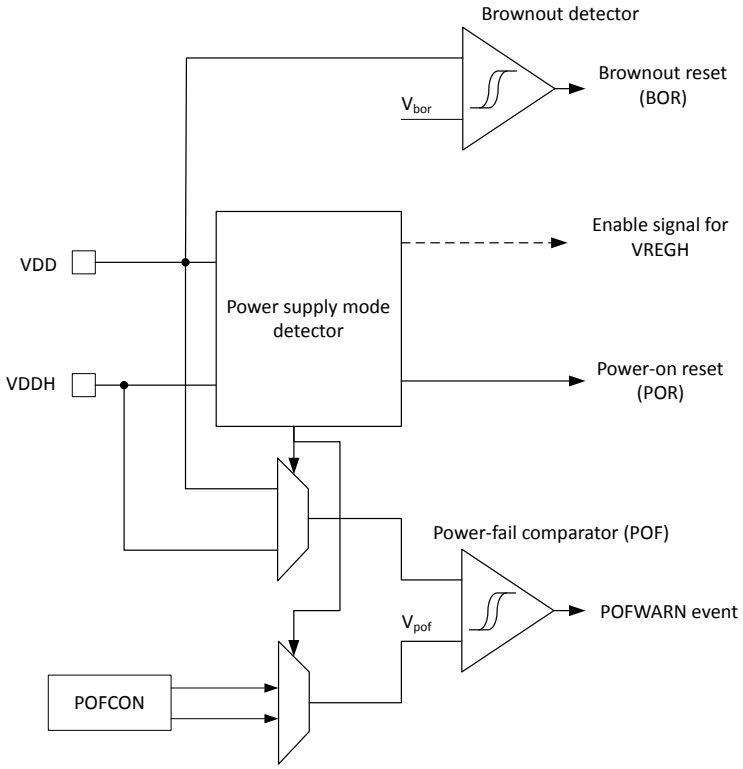

Figure 11: Power supply supervision in nRF5340

The power supply mode detector determines which supply pin is used when the device is powered up. It selects the power supply mode accordingly, generates the enable signal that automatically enables VREGH, and generates a power-on reset (POR) which initializes the device. For an overview of the different supply modes, see Power supply modes and regulators on page 38.

The brownout detector monitors the VDD supply (input of the VREGMAIN regulator) to ensure safe operation, and generates a brownout reset (BOR) if the voltage is too low, holding the device in reset when the voltage is too low for safe operation. The brownout reset voltage is defined in parameters $V_{BOR,OFF}$ and $V_{BOR,ON}$.

4.5.3.1 Power-fail comparator

The power-fail comparator (POF) can provide the CPU with an early warning of an impending power supply failure.

The POF can be used to signal the application when the supply voltage drops below a configured threshold. The POF will not reset the system, but instead give the CPU time to prepare for an orderly power-down. The following figure shows the main elements of the POF.

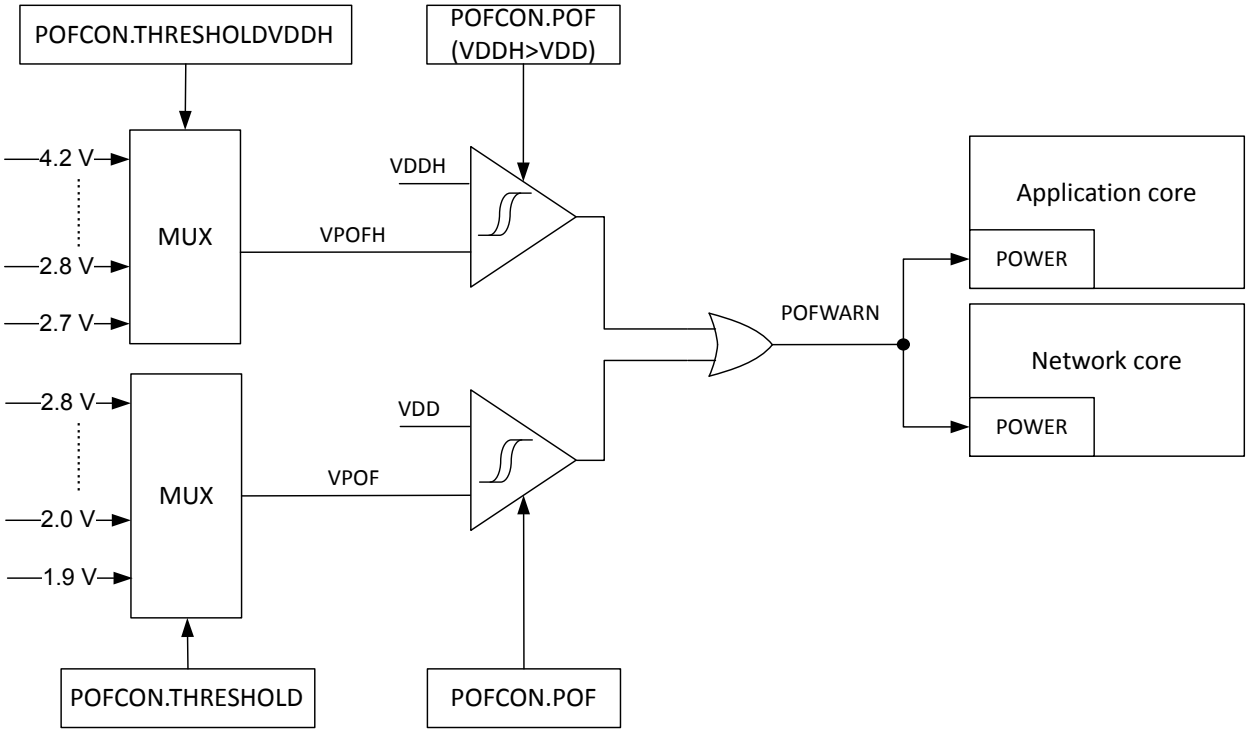

Figure 12: Power-fail comparator

Using the POF is optional, and must be enabled and configured through the register POFCON (Retained) on page 52.

Depending on the supply mode (see Power supply modes and regulators on page 38), the thresholds V_{POF} and V_{POFH} must be configured to a suitable level through the POFCON register. When the supply voltage falls below the defined threshold, the POF will generate the event POFWARN that is sent to the POWER module within both the application and network cores. Software running on the two cores can use this signal to prepare for a power failure. This event is also generated when the supply voltage is already below the threshold at the time the power-fail comparator is enabled, or if the threshold is reconfigured to a level above the supply voltage.

If the POF is enabled and the supply voltage is below the threshold, the POF will prevent the NVMC from performing write operations to the NVM.

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

The POF features a hysteresis of $V_{POFHYST}$, as illustrated in the following figure.

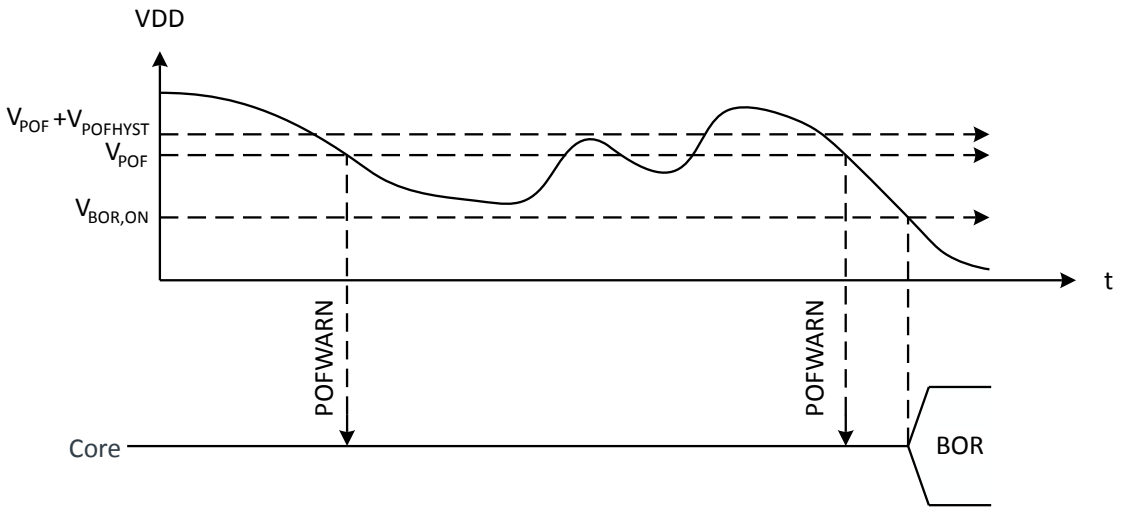

Figure 13: POF hysteresis and POFWARN event (BOR = brownout reset)

The POF hysteresis voltage is defined with the $V_{POFHYST}$ parameter in Electrical specification on page 53.

4.6 POWER — Power control

The POWER peripheral provides an interface for the power and clock subsystem for task, event, and interrupt related settings.

Each core has its own POWER peripheral that is responsible for requesting resources from the power and clock subsystem. The power and clock subsystem ensure that the power mode with the proper latency settings is selected when requested by an instance of the POWER peripheral. This means that for the core, the Constant latency mode is prioritized over Low-power mode. For an overview of power modes, see Power submodes on page 23.

The POFWARN event is a system level event that enables each core to react quickly if there is a power failure. The power-fail comparator must be configured and enabled in order to receive the event, see Power-fail comparator on page 41 for more information.

Power control of the RAM blocks is controlled by the Volatile memory controller (VMC), see VMC — Volatile memory controller on page 734.

Note: Registers INTEN on page 47 ,INTENSET on page 47, and INTENCLR on page 47 are shared between the POWER and CLOCK peripehrals.

4.6.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50005000 0x40005000	APPLICATION	POWER	POWER : S POWER : NS	US	NA	Power control	
0x41005000	NETWORK	POWER	POWER	NS	NA	Power control	

Table 6: Instances

Register	Offset	Security	Description
TASKS_CONSTLAT	0x78		Enable Constant Latency mode
TASKS_LOWPWR	0x7C		Enable Low-Power mode (variable latency)

Register	Offset	Security	Description	
SUBSCRIBE_CONSTLAT	0xF8		Subscribe configuration for task CONSTLAT	
SUBSCRIBE_LOWPWR	0xFC		Subscribe configuration for task LOWPWR	
EVENTS_POFWARN	0x108		Power failure warning	
EVENTS_SLEEPENTER	0x114		CPU entered WFI/WFE sleep	
EVENTS_SLEEPEXIT	0x118		CPU exited WFI/WFE sleep	
PUBLISH_POFWARN	0x188		Publish configuration for event POFWARN	
PUBLISH_SLEEPENTER	0x194		Publish configuration for event SLEEPENTER	
PUBLISH_SLEEPEXIT	0x198		Publish configuration for event SLEEPEXIT	
INTEN	0x300		Enable or disable interrupt	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
GPREGRET[n]	0x51C		General purpose retention register	Retained

Table 7: Register overview

4.6.1.1 TASKS_CONSTLAT

Address offset: 0x78

Enable Constant Latency mode

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_CONSTLAT			Enable Constant Latency mode
			Trigger	1	Trigger task

4.6.1.2 TASKS_LOWPWR

Address offset: 0x7C

Enable Low-Power mode (variable latency)

Bit number	31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value	Description
A	W	TASKS_LOWPWR			Enable Low-Power mode (variable latency)
			Trigger	1	Trigger task

4.6.1.3 SUBSCRIBE_CONSTLAT

Address offset: 0xF8

Subscribe configuration for task CONSTLAT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CONSTLAT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.6.1.4 SUBSCRIBE_LOWPWR

Address offset: 0xFC

Subscribe configuration for task LOWPWR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task LOWPWR will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.6.1.5 EVENTS_POFWARN

Address offset: 0x108

Power failure warning

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_POFWARN			Power failure warning
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.6.1.6 EVENTS_SLEEPENTER

Address offset: 0x114

CPU entered WFI/WFE sleep

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SLEEPENTER			CPU entered WFI/WFE sleep
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.6.1.7 EVENTS_SLEEPEXIT

Address offset: 0x118

CPU exited WFI/WFE sleep

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SLEEPEXIT			CPU exited WFI/WFE sleep
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.6.1.8 PUBLISH_POFWARN

Address offset: 0x188

Publish configuration for event POFWARN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event POFWARN will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.6.1.9 PUBLISH_SLEEPENTER

Address offset: 0x194

Publish configuration for event SLEEPENTER

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SLEEPENTER will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.6.1.10 PUBLISH_SLEEPEXIT

Address offset: 0x198

Publish configuration for event SLEEPEXIT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SLEEPEXIT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.6.1.11 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	POFWARN			Enable or disable interrupt for event POFWARN
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	SLEEPENTER			Enable or disable interrupt for event SLEEPENTER
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	SLEEPEXIT			Enable or disable interrupt for event SLEEPEXIT
			Disabled	0	Disable
			Enabled	1	Enable

4.6.1.12 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	POFWARN			Write '1' to enable interrupt for event POFWARN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	SLEEPENTER			Write '1' to enable interrupt for event SLEEPENTER
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	SLEEPEXIT			Write '1' to enable interrupt for event SLEEPEXIT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.6.1.13 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	POFWARN			Write '1' to disable interrupt for event POFWARN
			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				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	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	SLEEPENTER			Write '1' to disable interrupt for event SLEEPENTER
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	SLEEPEXIT			Write '1' to disable interrupt for event SLEEPEXIT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.6.1.14 GPREGRET[n] (n=0..1) (Retained)

Address offset: 0x51C + (n × 0x4)

This register is a retained register

General purpose retention register

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

4.7 REGULATORS — Regulator control

All system components are powered from the on-chip voltage regulators. These regulators are responsible for converting the voltage supplied on the VDD or VDDH pins to adequate voltages to be used internally.

The available regulators can be configured in multiple ways to accomodate different input voltage ranges. Some modes support sourcing power to external circuitry. The voltage modes that are supported by nRF5340 are listed in the following table.

Voltage mode	Input voltage range	Output voltage range
Normal voltage mode	1.7 V to 3.6 V	-
High voltage mode	2.5 V to 5.5 V	1.8 V to 3.3 V

Table 8: Supported voltage modes

For an overview on the available regulators, see Power supply modes and regulators on page 38.

4.7.1 Normal voltage mode

Normal voltage mode uses the main regulator (VREGMAIN) and the radio regulator (VREGRADIO).

The VREGMAIN and VREGRADIO regulators operate in LDO mode by default. DC/DC mode can be enabled independently for each regulator using VREGMAIN.DCDCEN (Retained) on page 52 and VREGRADIO.DCDCEN (Retained) on page 53 respectively.

When configured as shown in the following figure, nRF5340 enters normal voltage mode. Here both regulators are in DC/DC mode. An external LC filter is required for each regulator in DC/DC mode. If a regulator is only to be used in LDO mode, the inductor for this regulator is not needed. In this mode, the VDDH pin must be connected to VDD, even if the high voltage regulator (VREGH) is not in use.

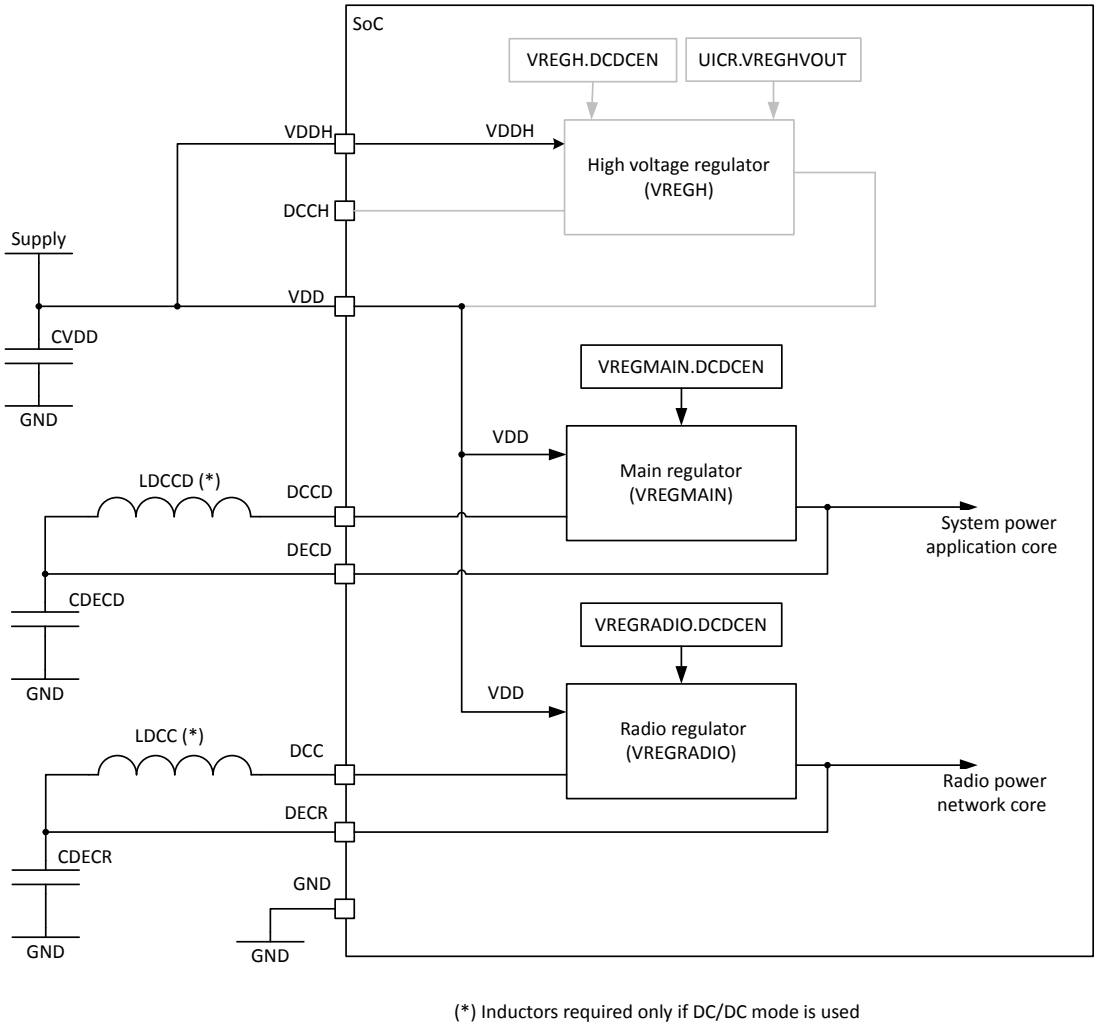

Figure 14: Normal voltage mode

The advantage of using a regulator in DC/DC mode is that the overall power consumption is reduced. This is because the regulator in DC/DC mode has a higher efficiency than in LDO mode. Regulator efficiency in DC/DC mode varies depending on the supply voltage and the current drawn from the regulators.

4.7.2 High voltage mode

High voltage mode uses the main regulator (VREGMAIN), the high voltage regulator (VREGH), and the radio regulator (VREGRADIO).

All regulators operate in LDO mode by default. DC/DC mode can be enabled independently for each regulator using VREGMAIN.DCDCEN (Retained) on page 52, VREGH.DCDCEN (Retained) on page 53, and VREGRADIO.DCDCEN (Retained) on page 53.

When configured as shown in the following figure, nRF5340 enters high voltage mode. Here all three regulators are in DC/DC mode. An external LC filter is required for each of the regulators in DC/DC mode. If a regulator is only to be used in LDO mode, the inductor for this regulator is not needed.

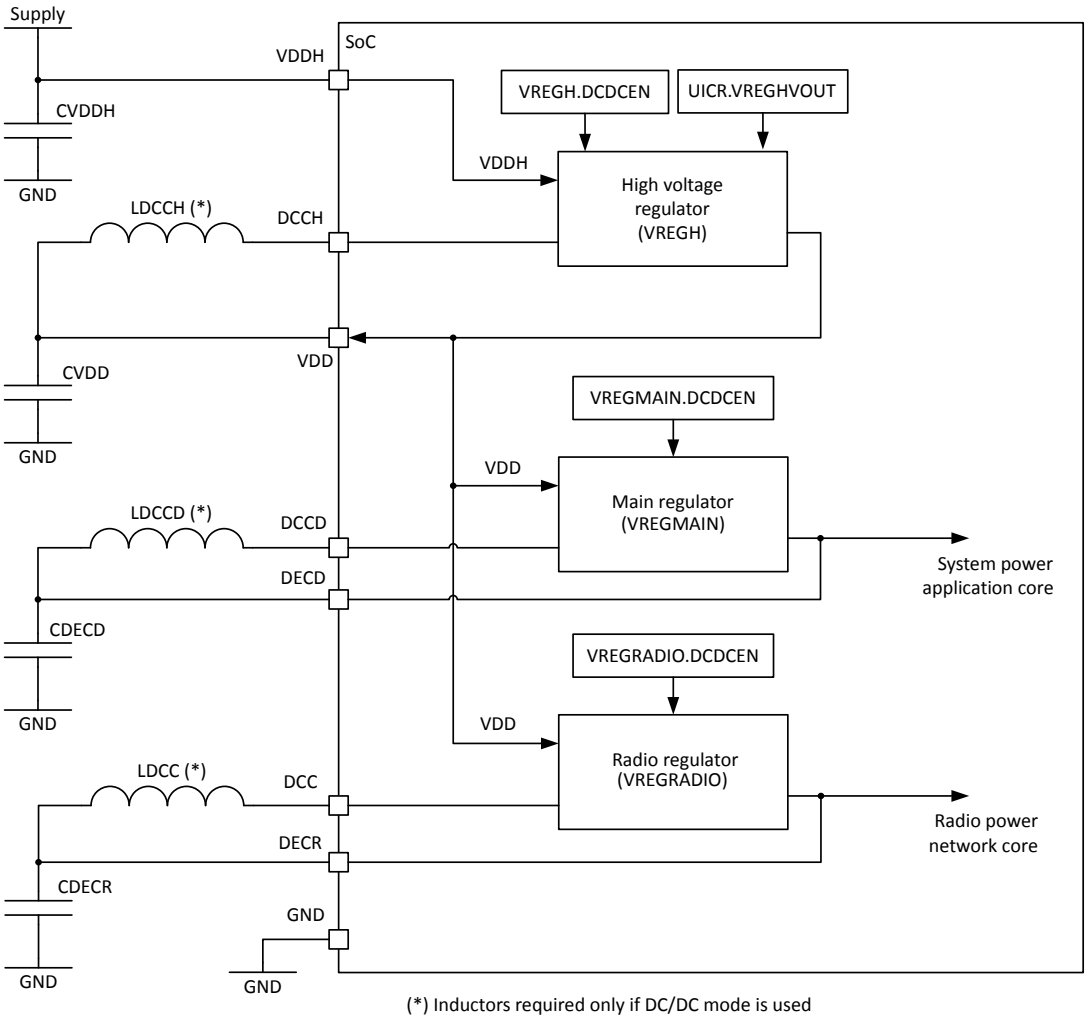

Figure 15: High voltage mode

The advantage of using a regulator in DC/DC mode is that the overall power consumption is reduced. This is because the regulator in DC/DC mode has higher efficiency than when in LDO mode. Regulator efficiency in DC/DC mode varies depending on the supply voltage and the current drawn from the regulators.

4.7.2.1 External circuitry supply

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

As illustrated in High voltage mode on page 49, external circuitry can be powered from the VDD pin. The VDD output voltage is programmed in the register UICR.VREGHVOUT.

The supported output voltage range depends on the supply voltage provided to the VDDH pin. The difference between voltage supplied on the VDDH pin and the voltage output on the VDD pin is defined by the $V_{REGH,DROP}$ parameter in Regulator specifications, VREGH stage on page 54.

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

Note: The maximum allowed current drawn by external circuitry is dependent on the total internal current draw. The maximum current that can be drawn externally from REGH is defined in Regulator specifications, VREGH stage on page 54).

4.7.3 GPIO levels

The GPIO high reference voltage depends on the regulator voltage mode.

In normal voltage mode, the GPIO high level equals the voltage supplied to the VDD pin. In high voltage mode, it equals the level specified in the VREGHVOUT register.

4.7.4 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50004000 0x40004000	APPLICATION	REGULATORS	REGULATORS : S REGULATORS : NS	US	NA	Regulator configuration	

Table 9: Instances

Register	Offset	Security	Description	
MAINREGSTATUS	0x428		Main supply status	Retained
SYSTEMOFF	0x500		System OFF register	
POFCON	0x510		Power-fail comparator configuration	Retained
VREGMAIN.DCDCEN	0x704		DC/DC enable register for VREGMAIN	Retained
VREGRADIO.DCDCEN	0x904		DC/DC enable register for VREGRADIO	Retained
VREGH.DCDCEN	0xB00		DC/DC enable register for VREGH	Retained

Table 10: Register overview

4.7.4.1 MAINREGSTATUS (Retained)

Address offset: 0x428

This register is a retained register

Main supply status

Note:

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	R	VREGH			VREGH status
			Inactive	0	Normal voltage mode. Voltage supplied on VDD and VDDH.
			Active	1	High voltage mode. Voltage supplied on VDDH.

4.7.4.2 SYSTEMOFF

Address offset: 0x500

System OFF register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	SYSTEMOFF			Enable System OFF mode
			Enter	1	Enable System OFF mode

4.7.4.3 POFCON (Retained)

Address offset: 0x510

This register is a retained register

Power-fail comparator configuration

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					D D D D B B B B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	POF			Enable or disable power-fail comparator
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	THRESHOLD			Power-fail comparator threshold setting
			V19	6	Set threshold to 1.9 V
			V20	7	Set threshold to 2.0 V
			V21	8	Set threshold to 2.1 V
			V22	9	Set threshold to 2.2 V
			V23	10	Set threshold to 2.3 V
			V24	11	Set threshold to 2.4 V
			V25	12	Set threshold to 2.5 V
			V26	13	Set threshold to 2.6 V
			V27	14	Set threshold to 2.7 V
			V28	15	Set threshold to 2.8 V
D	RW	THRESHOLDVDDH			Power-fail comparator threshold setting for voltage supply on VDDH
			V27	0	Set threshold to 2.7 V
			V28	1	Set threshold to 2.8 V
			V29	2	Set threshold to 2.9 V
			V30	3	Set threshold to 3.0 V
			V31	4	Set threshold to 3.1 V
			V32	5	Set threshold to 3.2 V
			V33	6	Set threshold to 3.3 V
			V34	7	Set threshold to 3.4 V
			V35	8	Set threshold to 3.5 V
			V36	9	Set threshold to 3.6 V
			V37	10	Set threshold to 3.7 V
			V38	11	Set threshold to 3.8 V
			V39	12	Set threshold to 3.9 V
			V40	13	Set threshold to 4.0 V
			V41	14	Set threshold to 4.1 V
			V42	15	Set threshold to 4.2 V

4.7.4.4 VREGMAIN.DCDCEN (Retained)

Address offset: 0x704

This register is a retained register

DC/DC enable register for VREGMAIN

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	DCDCEN			Enable or disable DC/DC converter
			Disabled	0	Disable
			Enabled	1	Enable

4.7.4.5 VREGRADIO.DCDCEN (Retained)

Address offset: 0x904

This register is a retained register

DC/DC enable register for VREGRADIO

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	DCDCEN			Enable or disable DC/DC converter
			Disabled	0	Disable
			Enabled	1	Enable

4.7.4.6 VREGH.DCDCEN (Retained)

Address offset: 0xB00

This register is a retained register

DC/DC enable register for VREGH

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	DCDCEN			Enable or disable DC/DC converter
			Disabled	0	Disable
			Enabled	1	Enable

4.7.5 Electrical specification

4.7.5.1 Recommended operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
$V_{DD,POR}$	VDD supply voltage needed during power-on reset.	1.75			V
V_{DD}	Normal voltage mode operating voltage.	1.7		3.6	V
V_{DDH}	High voltage mode operating voltage.	2.5		5.5	V

4.7.5.2 Regulator specifications, VREGH stage

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDOUT}	VDD output voltage.	1.8		3.3	V
$V_{DDOUT,ERR}$	VDD output voltage error (deviation from setting in).	-10		+5	%
$I_{EXT,OFF}$	External current draw[1] allowed in high voltage mode (supply on VDDH) during System OFF.			1	mA
$I_{EXT,DCDC}$	External current draw[1] allowed in High Voltage mode (supply on VDDH) when VREGMAIN and VREGRADIO are in DC/DC mode. Assumes worst-case power consumption from both cores[2], and at the lowest VDD output voltage setting.			7	mA
$I_{EXT,LDO}$	External current draw[1] allowed in High Voltage mode (supply on VDDH) when when VREGMAIN and VREGRADIO are in LDO mode. Assumes worst-case power consumption from both cores[2], and at the lowest VDD output voltage setting.			1	mA
$V_{REGH,DROP}$	Required difference between input voltage (VDDH) and output voltage (VDD, configured in VREGHVOUT on page 125), VDDH > VDD			0.3	V

4.7.5.3 Regulator startup times

Symbol	Description	Min.	Typ.	Max.	Units
$t_{RISE,VREGHOUT}$	VREGH output (VDD) rise time after VDDH reaches minimum VDDH supply voltage	..	..	..	
$t_{RISE,VREGHOUT,10us}$	VDDH rise time 10 µs		0.2	1.6	ms
$t_{RISE,VREGHOUT,10ms}$	VDDH rise time 10 ms		5		ms
$t_{RISE,VREGHOUT,50ms}$	VDDH rise time 50 ms	30	50	80	ms

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

[2] In practice, the maximum external current draw is limited by the maximum output current of VREGH, subtracting the actual current being drawn from VDD.

4.7.5.4 Power-fail comparator

Symbol	Description	Min.	Typ.	Max.	Units
$V_{POF,NV}$	Nominal power level warning thresholds (falling supply voltage) in normal voltage mode (supply on VDD). Levels are configurable between min. and max. in increments of 100 mV.		1.7	2.8	V
$V_{POF,HV}$	Nominal power level warning thresholds (falling supply voltage) in high Voltage mode (supply on VDDH). Levels are configurable between min. and max. in increments of 100 mV.		2.7	4.2	V
V_{POFTOL}	Threshold voltage tolerance (applies in both normal voltage mode and high voltage mode).	-5		+5	%
$V_{POFHYST}$	Threshold voltage hysteresis (applies in both normal voltage mode and high voltage mode).	40	50	60	mV
$V_{BOR,OFF}$	Brownout reset voltage range System OFF mode. Brownout only applies to the voltage on VDD.	1.54		1.64	V
$V_{BOR,ON}$	Brownout reset voltage range System ON mode. Brownout only applies to the voltage on VDD.	1.57		1.63	V

4.8 USBREG — USB regulator control

The USB peripheral has its own voltage regulator. When using the USB peripheral, a 5 V USB supply needs to be provided on the VBUS pin.

The USB peripheral has a dedicated internal voltage regulator for converting the VBUS supply to 3.3 V to be used by the USB signalling interface (D+ and D- lines, and pull-up on D+). The rest of the USB peripheral (USBD) is supplied through the main supply like other on-chip features. As a consequence, both VBUS and combinations of VDDH and VDD are required for USB peripheral operation. For details on configuring the main supplies, see Power supply modes and regulators on page 38.

When VBUS rises into its valid range, the software is notified through the USBDETECTED event. The USBREMOVED event is sent when VBUS goes below its valid range. Use these events to implement the USBD startup sequence described in USBD power-up sequence on page 693.

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

See VBUS detection specifications on page 60 for the voltage level where events are sent ($V_{BUS,DETECT}$ and $V_{BUS,REMOVE}$) or where the system causes a wakeup from System OFF ($V_{BUS,DETECT}$).

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

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

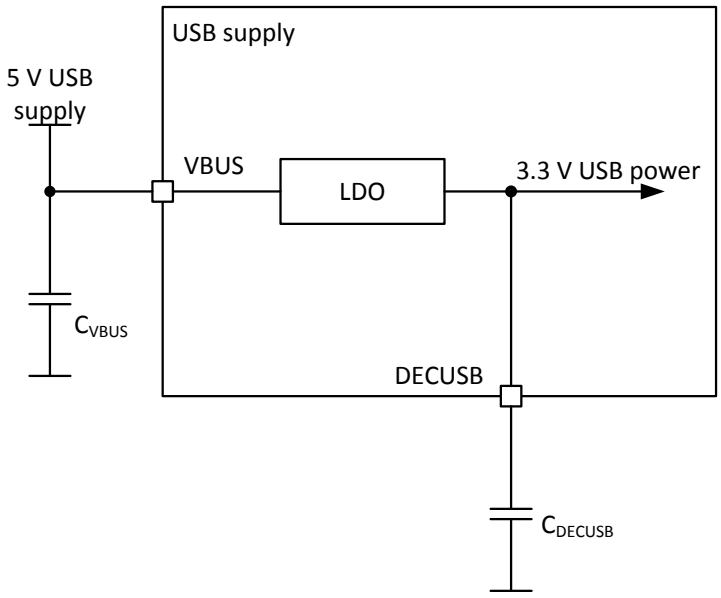

Figure 16: USB voltage regulator

To ensure stability, the input and output of the USB regulator need to be decoupled with a suitable decoupling capacitor C_{VBUS}. See Reference circuitry on page 788 for the recommended values.

4.8.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50037000 0x40037000	APPLICATION	USBREG	USBREGULATOR : S USBREGULATOR : NS	US	NA	USB regulator control	

Table 11: Instances

Register	Offset	Security	Description
EVENTS_USBDETECTED	0x100		Voltage supply detected on VBUS
EVENTS_USBREMOVED	0x104		Voltage supply removed from VBUS
EVENTS_USBPWRRDY	0x108		USB 3.3 V supply ready
PUBLISH_USBDETECTED	0x180		Publish configuration for event USBDETECTED
PUBLISH_USBREMOVED	0x184		Publish configuration for event USBREMOVED
PUBLISH_USBPWRRDY	0x188		Publish configuration for event USBPWRRDY
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
USBREGSTATUS	0x400		USB supply status

Table 12: Register overview

4.8.1.1 EVENTS_USBDETECTED

Address offset: 0x100

Voltage supply detected on VBUS

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_USBDETECTED			Voltage supply detected on VBUS
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.8.1.2 EVENTS_USBREMOVED

Address offset: 0x104

Voltage supply removed from VBUS

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_USBREMOVED			Voltage supply removed from VBUS
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.8.1.3 EVENTS_USBPWRRDY

Address offset: 0x108

USB 3.3 V supply ready

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_USBPWRRDY			USB 3.3 V supply ready
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.8.1.4 PUBLISH_USBDETECTED

Address offset: 0x180

Publish configuration for event USBDETECTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event USBDETECTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.8.1.5 PUBLISH_USBREMOVED

Address offset: 0x184

Publish configuration for event USBREMOVED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event USBREMOVED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.8.1.6 PUBLISH_USBPWRRDY

Address offset: 0x188

Publish configuration for event USBPWRRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event USBPWRRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.8.1.7 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	USBDETECTED			Enable or disable interrupt for event USBDETECTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	USBREMOVED			Enable or disable interrupt for event USBREMOVED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	USBPWRRDY			Enable or disable interrupt for event USBPWRRDY
			Disabled	0	Disable
			Enabled	1	Enable

4.8.1.8 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	USBDETECTED			Write '1' to enable interrupt for event USBDETECTED

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

4.8.1.9 INTENCLR

Address offset: 0x308

Disable interrupt

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

4.8.1.10 USBREGSTATUS

Address offset: 0x400

USB supply status

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

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

4.8.2 Electrical specification

4.8.2.1 USB operating conditions

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

4.8.2.2 USB regulator specifications

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,QUIES}$	USB regulator quiescent current drawn from VBUS (USBD enabled)		170		µA
$t_{USBPWRRDY}$	Time from USB enabled to USBPWRRDY event triggered, V_{BUS} supply provided		1		ms
V_{USB33}	On voltage at the USB regulator output (DECUSB pin)	3.0	3.3	3.6	V
$R_{SOURCE,VBUS}$	Maximum source resistance on VBUS, including cable			6[3]	Ω
C_{DECUSB}	Decoupling capacitor on the DECUSB pin	2.35	4.7	5.5	µF

4.8.2.3 VBUS detection specifications

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

4.9 VREQCTRL — Voltage request control

The VREQCTRL can request additional voltage on the VREGRADIO regulated supply to support +3 dBm TX power on RADIO.

Setting the VREGRADIO.VREQH register will request high voltage. The request is active until the register is cleared. Status register VREGRADIO.VREQHREADY indicates when the regulator has changed to high voltage.

[3] When VDDH is not connected to VBUS.

4.9.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x41004000	NETWORK	VREQCTRL	VREQCTRL	NS	NA	Voltage request control	

Table 13: Instances

Register	Offset	Security	Description	
VREGRADIO.VREQH	0x500		Request high voltage on RADIO After requesting high voltage, the user must wait until VREQHREADY is set to Ready	Retained
VREGRADIO.VREQHREADY	0x508		High voltage on RADIO is ready	

Table 14: Register overview

4.9.1.1 VREGRADIO.VREQH (Retained)

Address offset: 0x500

This register is a retained register

Request high voltage on RADIO

After requesting high voltage, the user must wait until VREQHREADY is set to Ready

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	VREQH			Request high voltage
			Disabled	0	Disable
			Enabled	1	Enable

4.9.1.2 VREGRADIO.VREQHREADY

Address offset: 0x508

High voltage on RADIO is ready

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	R	READY			RADIO is ready to operate on high voltage
			NotReady	0	Not ready
			Ready	1	Ready

4.10 RESET — Reset control

A reset in the system is triggered by either a system-level or core-level reset source.

A system-level reset will reset all cores. Power-on reset, brownout reset, and pin reset are examples of a system-level reset. A core-level reset, such as a soft reset or a lockup, will reset either the entire core or only part of it. The different reset sources in the system are illustrated in the following figure.

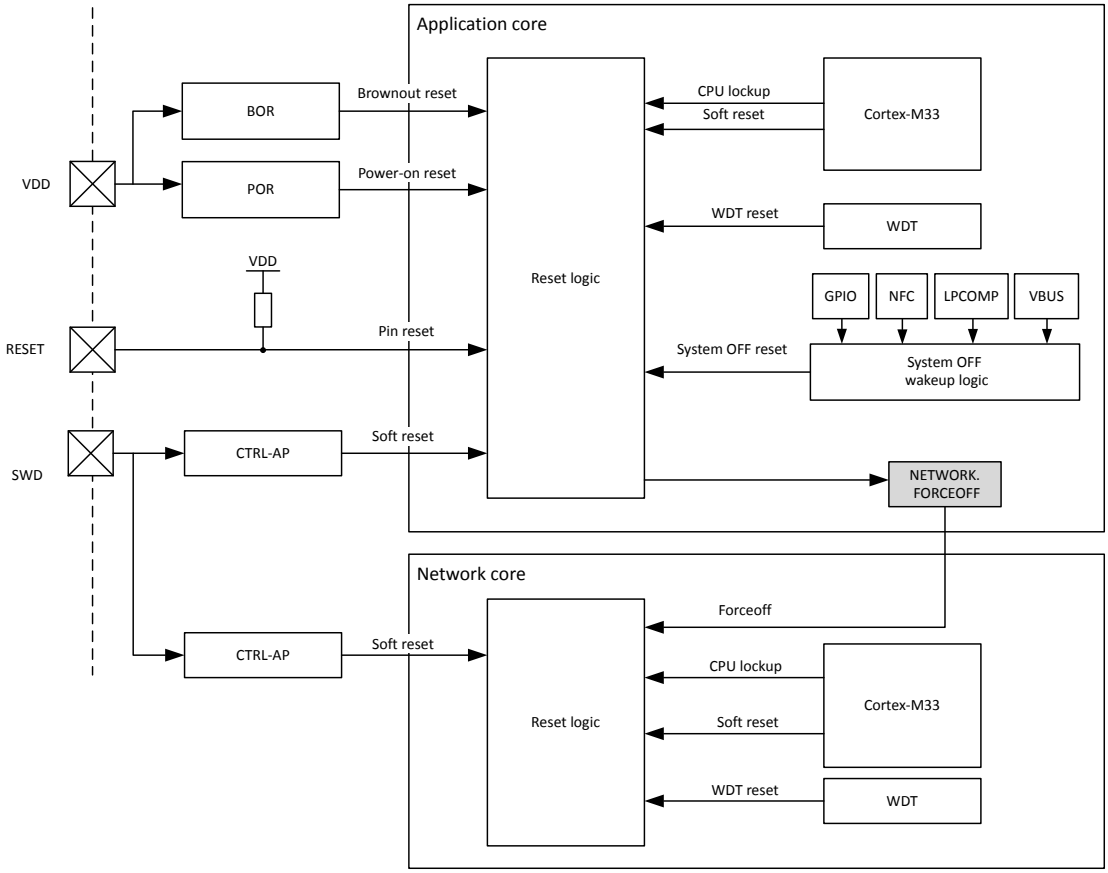

Figure 17: Reset sources

After a system-level reset, the application core will start up on its own. The network core is not automatically started, but can be started by the application core CPU, see Network Force-OFF on page 63.

After a reset has occurred, the register RESETREAS on page 66 can be read to determine which source generated the reset. Each core has its own RESETREAS register. System-level and application core reset sources are also available in the network core's RESETREAS register, unless otherwise noted.

4.10.1 Power-on reset

The power-on reset (POR) generator initializes the system when the VDD supply voltage is above the power-on threshold. This also applies in high voltage mode, where the VDD supply voltage is provided by the high voltage regulator (VREGH).

The system is held in a reset state until the supply has reached the minimum operating voltage, and the internal voltage regulators have started. After a power-on reset, the application core is started while the network core is held in reset, see Network Force-OFF on page 63.

4.10.2 Pin reset

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

Similar to a power-on reset, the application core is started after the reset pin is deasserted. The network core is held in reset, see Network Force-OFF on page 63.

The reset pin has an internal pull-up resistor with the same resistance as GPIO pull-ups, see GPIO — General purpose input/output on page 219.

4.10.3 Brownout reset

The brownout reset (BOR) generator puts the system in reset state if the VDD supply voltage drops below the brownout reset threshold. This also applies in high voltage mode, where the VDD supply voltage is provided by the high voltage regulator (VREGH).

Similar to a power-on reset, the application core is started after BOR is deasserted while the network core is held in reset, see Network Force-OFF on page 63.

4.10.4 Wakeup from System OFF mode reset

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

Similar to a power-on reset, the application core is started while the network core is held in reset, see Network Force-OFF on page 63.

If the device is in Debug Interface mode, the debug acces port (DAP) is not reset after a wakeup from System OFF. For more information, see Debug and trace on page 745.

4.10.5 Soft reset

Soft reset is generated when the SYSRESETREQ bit of the application interrupt and reset control register (AIRCR) in the Arm core of the application and network cores is set. For more information, see Arm documentation.

When the application core performs a soft reset, the network core is held in reset, see Network Force-OFF on page 63. A soft reset in the network core will only cause the network core to reset.

A soft reset can also be generated using the RESET on page 761 register in the associated CTRL-AP.

4.10.6 Watchdog timer reset

A watchdog timer (WDT) reset is generated when the watchdog timer times out.

Each core has its own WDT instance. When the application core gets a WDT reset, the network core is held in reset, see Network Force-OFF on page 63. A WDT reset in the network core will only cause the network core to reset. The reset target depends on the core where WDT is instantiated.

> **Note:** Because the network core WDT reset is local for the network core, the application core is not aware of WDT timing out in the network core. Notifying the application core is possible. One way is to check the register RESETREAS on page 66 for WDT flags and report the error through inter-processor communication (IPC).

For more information about WDT, see WDT — Watchdog timer on page 737. More information about IPC is available in IPC — Interprocessor communication on page 264.

4.10.7 Network Force-OFF

The application core can force the network core off.

The application core can hold the network core in Force-OFF mode, using register NETWORK.FORCEOFF on page 68.

Application core resets implicitly result in the network core being held in Force-OFF. The network core will be held in Force-OFF until the application core releases it using the NETWORK.FORCEOFF register.

For details on how to use this mode, see Force-OFF mode on page 24.

4.10.8 Retained registers

A retained register is one that retains its value in System OFF and/or Force-OFF modes and when reset, depending on the reset source. See individual peripheral chapters for information about which registers are retained for the various peripherals.

4.10.9 Application core reset behavior

Application core reset behavior depends on the reset source.

Any reset in the application core will cause a network core Force-OFF, triggering the FORCEOFF reset source in the network core. For more information, see Network Force-OFF on page 63.

In System OFF mode, the watchdog timer is not running and there is no CPU lockup possible. RAM may be fully or partially retained, depending on RAM retention settings in VMC — Volatile memory controller on page 734.

If the device is in Debug Interface mode, the debug components will not be reset. Additionally, CPU lockup will not generate a reset. See Debug and trace on page 745 for more information about the different debug components in the system.

Application core reset targets and their reset sources are summarized in the following table.

An 'x' in the table means that the specific module is reset.

Reset source	Reset target					
	CPU	Network core	Debug	RAM	WDT	RESETREAS
CPU lockup	x	x				
Soft reset	x	x				
Wakeup from System OFF mode reset	x	x	x	x[4]	x	
Watchdog timer reset	x	x	x	x	x	
Pin reset	x	x	x	x	x	
Brownout reset	x	x	x	x	x	x
Power-on reset	x	x	x	x	x	x
NETWORK.FORCEOFF		x				

Table 15: Application core reset targets and their reset sources

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

Some retained registers may have a different reset behavior, as shown in the following table.

[4] Depending on RAM retention settings.

An 'x' in the table means that the specific module is reset.

Reset source	Reset target				
	Regular peripheral registers	SPU	GPIO	REGULATORS, OSCILLATORS	POWER.GPREGRET
CPU lockup	x	x	x[5]		
Soft reset	x	x	x[5]		
Wakeup from System OFF mode reset	x				
Watchdog timer reset	x	x	x	x	
Pin reset	x	x	x	x	
Brownout reset	x	x	x	x	x
Power-on reset	x	x	x	x	x

Table 16: Application core reset behavior for retained registers

4.10.10 Network core reset behavior

Network core reset behavior depends on the reset source.

In System OFF mode, or when the network core is held in Force-OFF, the watchdog timer is not running and there is no CPU lockup possible. RAM may be fully or partially retained, depending on RAM retention settings in VMC — Volatile memory controller on page 734.

If the device is in Debug Interface mode, the debug components will not be reset. Additionally, CPU lockup will not generate a reset. See Debug and trace on page 745 for more information about the different debug components in the system.

Any reset in the application core will cause a network core force off, triggering the network FORCEOFF reset source in the following table. For more information, see Network Force-OFF on page 63.

An 'x' in the table means that the specific module is reset. Pin reset, brownout reset, and power-on reset are system level reset sources with the network core and application core having the same behavior, see Application core reset behavior on page 64.

Reset source	Reset target			
	CPU	RAM	WDT	RESETREAS
CPU lockup	x			
Soft reset	x			
Network FORCEOFF	x	x[6]	x	
Application Watchdog timer reset	x	x	x	
Local Watchdog timer reset	x	x	x	

Table 17: Network core reset target sources

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

Some retained registers may have a different reset behavior, as shown in following table.

[5] Except MCUSEL field, the MCUSEL register of the GPIO peripheral is not reset for CPU lockup and Soft reset.

[6] Depending on RAM retention settings.

An 'x' in the table means that the specific module is reset. Pin reset, brownout reset, and power-on reset are system level reset sources with the network core and application core having the same behavior, see Application core reset behavior on page 64.

Reset source	Reset target		
	Regular peripheral registers	GPIO	POWER.GPREGRET
CPU lockup	x	x [7]	
Soft reset	x	x [7]	
Network FORCEOFF	x		
Application Watchdog timer reset	x	x	
Local Watchdog timer reset	x	x	

Table 18: Network core reset behavior for retained registers

4.10.11 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50005000 0x40005000	APPLICATION	RESET	RESET : S RESET : NS	US	NA	Reset control and status	Not supported: LSREQ, LLOCKUP, LDOG, MRST, MFORCEOFF, LCTRLAP
0x41005000	NETWORK	RESET	RESET	NS	NA	Reset status	Not supported: NETWORK.FORCEOFF

Table 19: Instances

Register	Offset	Security	Description
RESETREAS	0x400		Reset reason
NETWORK.FORCEOFF	0x614		Force network core off

Table 20: Register overview

4.10.11.1 RESETREAS

Address offset: 0x400

Reset reason

Unless cleared, the RESETREAS register will be cumulative. A field is cleared by writing `1` to it. Multiple fields can be cleared at the same time by writing a value with several of the fields set to `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					Q	P	O	N	M					K	J	I									H	G	F	E	D	C	B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	RESETPIN			Reset from pin reset detected
			NotDetected	0	Not detected
			Detected	1	Detected
B	RW	DOG0			Reset from application watchdog timer 0 detected
			NotDetected	0	Not detected
			Detected	1	Detected
C	RW	CTRLAP			Reset from application CTRL-AP detected
			NotDetected	0	Not detected

[7] MCUSEL settings are kept.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Q P O N M K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Detected	1	Detected
D	RW	SREQ			Reset from application soft reset detected
			NotDetected	0	Not detected
			Detected	1	Detected
E	RW	LOCKUP			Reset from application CPU lockup detected
			NotDetected	0	Not detected
			Detected	1	Detected
F	RW	OFF			Reset due to wakeup from System OFF mode when wakeup is triggered by DETECT signal from GPIO
			NotDetected	0	Not detected
			Detected	1	Detected
G	RW	LPCOMP			Reset due to wakeup from System OFF mode when wakeup is triggered by ANADETECT signal from LPCOMP
			NotDetected	0	Not detected
			Detected	1	Detected
H	RW	DIF			Reset due to wakeup from System OFF mode when wakeup is triggered by entering the Debug Interface mode
			NotDetected	0	Not detected
			Detected	1	Detected
I	RW	LSREQ			Reset from network soft reset detected Function present only in network core
			NotDetected	0	Not detected
			Detected	1	Detected
J	RW	LLOCKUP			Reset from network CPU lockup detected Function present only in network core
			NotDetected	0	Not detected
			Detected	1	Detected
K	RW	LDOG			Reset from network watchdog timer detected Function present only in network core
			NotDetected	0	Not detected
			Detected	1	Detected
M	RW	MFORCEOFF			Force-OFF reset from application core detected Function present only in network core
			NotDetected	0	Not detected
			Detected	1	Detected
N	RW	NFC			Reset after wakeup from System OFF mode due to NFC field being detected
			NotDetected	0	Not detected
			Detected	1	Detected
O	RW	DOG1			Reset from application watchdog timer 1 detected
			NotDetected	0	Not detected
			Detected	1	Detected
P	RW	VBUS			Reset after wakeup from System OFF mode due to VBUS rising into valid range
			NotDetected	0	Not detected
			Detected	1	Detected
Q	RW	LCTRLAP			Reset from network CTRL-AP detected Function present only in network core
			NotDetected	0	Not detected

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

4.10.11.2 NETWORK.FORCEOFF

Address offset: 0x614

Force network core off

Function present only in application core

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	FORCEOFF			Force network core off
			Release	0	Release Force-OFF
			Hold	1	Hold Force-OFF

4.10.12 Electrical specification

4.10.12.1 Application core startup times

Symbol	Description	Min.	Typ.	Max.	Units
t_{POR}	Time in power-on reset after supply reaches minimum operating voltage, depending on supply rise time.	..	..	..	
$t_{POR,10us}$	VDD rise time 10 µs		0.7	1.0	ms
$t_{POR,10ms}$	VDD rise time > 10 ms		0		ms
t_{PINR}	Reset time when using pin reset, depending on pin capacitance	..	..	..	
$t_{PINR,500nF}$	500 nF capacitance at reset pin		13	40	ms
$t_{PINR,10uF}$	10 µF capacitance at reset pin		260	800	ms
t_{R2ON}	Time from reset to ON (CPU execute)		t_{POR} + t_{PINR}		
$t_{OFF2ON,NM}$	Time from OFF to CPU execute when in normal voltage mode (supply on VDD)				µs
$t_{OFF2ON,LDO,HV}$	Time from OFF to CPU execute when in high voltage mode (supply on VDDH) and VREGH using LDO regulator				µs
$t_{OFF2ON,DCDC,HV}$	Time from OFF to CPU execute when in high voltage mode (supply on VDDH) and VREGH using DC/DC regulator				µs
$t_{IDLE2CPU}$	Time from IDLE to CPU execute				µs
$t_{EVTSET,CL1}$	Time from HW event to PPI event in Constant Latency System ON mode		62.5		ns
$t_{EVTSET,CL0}$	Time from HW event to PPI event in Low-Power System ON mode		62.5		ns

4.10.12.2 Network core startup times

Symbol	Description	Min.	Typ.	Max.	Units
$t_{NET,EVTSET,CL1}$	Time from HW event to PPI event in Constant Latency System ON mode		62.5		ns
$t_{NET,EVTSET,CL0}$	Time from HW event to PPI event in Low-Power System ON mode		62.5		ns
$t_{NET,IDLE2CPU}$	Time from IDLE to CPU execute				µs
$t_{FO2ON,NET64}$	Time for network core from OFF to CPU execute after NETWORK.FORCEOFF is released				µs

4.11 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 peripherals and modules based on their individual requirements. Clock distribution is automated and grouped independently by module to limit current consumption in unused branches of the clock tree.

Each core subsystem has its own clock control system that is responsible for requesting resources from the power and clock subsystem.

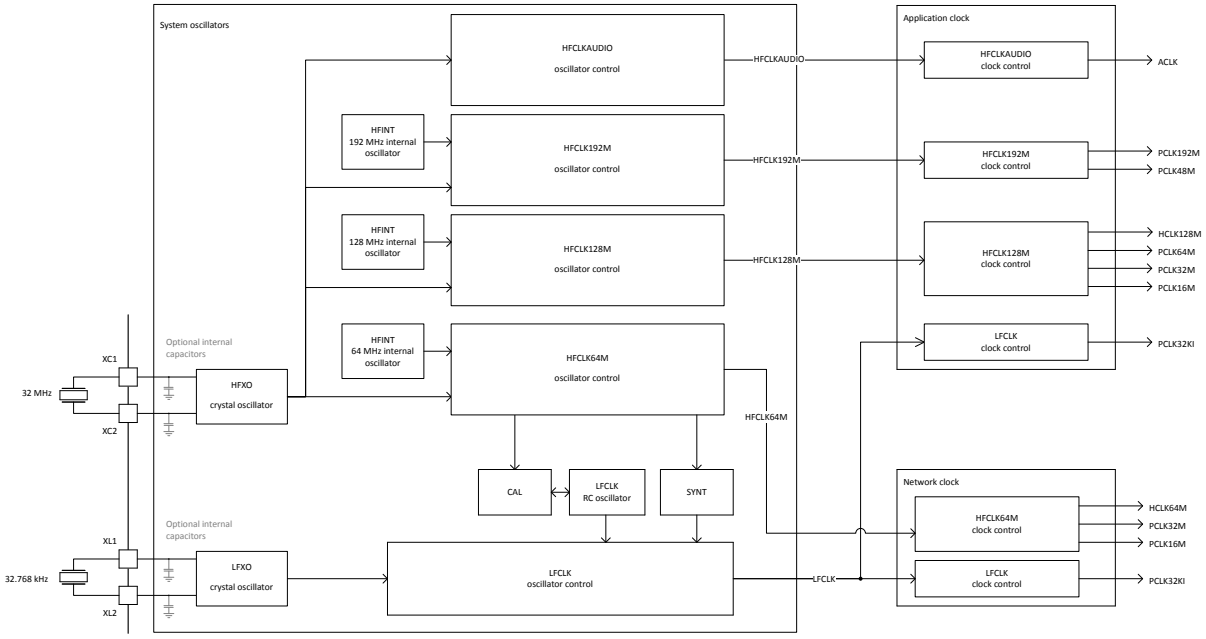

Figure 18: Clock control

The power and clock subsystem secures glitch-free switching from one clock source to another. This applies to all clock sources.

Note: Registers INTEN on page 84, INTENSET on page 85, and INTENCLR on page 85 are the same registers (at the same address) as the corresponding registers in POWER.

4.11.1 HFCLK controller

Each core has a number of high frequency clock (HFCLK) control instances. Each instance distributes one or more clocks to the core.

The following table lists the core clocks that are available.

Core clock	Description
HCLK128M	Scalable 128 MHz CPU clock for the application core
HCLK64M	64 MHz CPU clock for the network core
PCLK192M	Scalable 192 MHz clock for QSPI
PCLK64M	64 MHz peripheral clock
PCLK48M	48 MHz clock for USB
PCLK32M	32 MHz peripheral clock
PCLK16M	16 MHz peripheral clock
ACLK	11.289 MHz or 12.288 MHz tunable audio peripheral clock

Table 21: Core clocks

The HFCLK clocks sourced from the power and clock subsystem to the HFCLK control instances are the following:

HFCLK clock	Description
HFCLK128M	128 MHz HFCLK clock
HFCLK64M	64 MHz HFCLK clock
HFCLK192M	192 MHz HFCLK clock
HFCLKAUDIO	Audio HFCLK clock

Table 22: HFCLK clocks for HFCLK control instances

In order to generate the HFCLK clocks, the following HFCLK sources are available:

- 192 MHz/128 MHz/64 MHz internal oscillator (HFINT)
- 32 MHz crystal oscillator (HFXO), optionally using built-in capacitors as described in OSCILLATORS — Oscillator control on page 94

See Clock control on page 69 for more information.

CPUs, peripherals, and other system components within a core will automatically request clocks from its corresponding local HFCLK control. The HFCLK control passes the request to the power and clock subsystem and, once the clocks are running, distributes them to the components within the core.

When HFCLK control requests within a core are stopped, the HFCLK control will stop requesting clock from the power and clock subsystem. For example, when the CPU enters sleep or when peripherals have completed their tasks. If there are no HFCLK control requests from any core, the power and clock subsystem will automatically stop the clock.

When the system enters System ON mode, and a HFCLK clock is requested, the relevant HFINT will be used as the HFCLK source. When requests for the clock are stopped, the HFINT will automatically stop.

HFCLK clocks are only available to the HFCLK controllers when the system is in System ON mode.

It is possible to have a HFCLK source running before being started by the relevant clock request (for instance, the HFCLK source is kept running during sleep). This gives shorter start-up time but causes increased power consumption. Starting the HFXO is needed when crystal clock accuracy is required.

The HFCLK source selected in register HFCLKSRC on page 90 is started by triggering the HFCLKSTART task.

The source for the HFCLK128M/HFCLK64M clocks can be configured at any time (for instance, when the HFCLK has already been started). The content of the HFCLKSRC register only takes effect when the HFCLKSTART task is triggered.

The event HFCLKSTARTED is generated when the HFCLKSTART task is triggered, the oscillator is started, and the frequency is stabilized.

The HFCLK source selected in register HFCLK192MSRC on page 92 is started by triggering the HFCLK192MSTART task.

The source for the HFCLK192M clock can be configured at any time (for instance, when the HFCLK has already been started). The content of the HFCLK192MSRC register only takes effect when the HFCLK192MSTART task is triggered.

The event HFCLK192MSTARTED is generated when the HFCLK192MSTART task is triggered, the oscillator is started, and the frequency stabilized.

HFCLKAUDIO requires HFXO, so when triggering the HFCLKAUDIOSTART task, this always starts the HFXO.

The event HFCLKAUDIOSTARTED is generated when the HFCLKAUDIOSTART task is triggered, the oscillator is started, and the frequency stabilized.

It is possible to trigger a new START task after one has already been triggered, and before the corresponding STARTED event is generated. In this case, only one STARTED event will be generated, corresponding to the last triggered START task. Triggering a START task after the STARTED event from a previous triggered START taks is generated, will generate a new STARTED event.

Time from a START task to the corresponding STARTED event may differ depending on whether the HFCLK source is already running or in the process of starting. The amount of time before a STARTED event may vary when a different HFCLK source is configured before triggering a new START task.

When the clock control system switches from HFINT source to HFXO source, the HFXO becomes active. The startup time is programmable, enabling the use of different types of crystal oscillators (e.g. standard crystals that may have different startup times). The HFXO startup time is given as the sum of the following:

- HFXO power-up time, as specified in 32 MHz crystal oscillator (HFXO) on page 98.
- HFXO count time, as specified in HFXOCNT on page 126.

The HFXO must be selected and started in order to do the following:

- Use RADIO
  - The network domain HFCLKSTART task is used
- Enable USBD to respond to USB traffic
  - The application domain HFCLK192MSTART is used
- Set NFCT to activated state
  - The application domain HFCLKSTART task is used
- Improve SAADC performance by reducing clock jitter
  - The application domain HFCLKSTART task is used

Each HFCLK control can request the HFXO source independently from one another via the corresponding START task. This ensures that each core and peripheral will have access to a high accuracy clock when needed. Core clocks that originate from the same HFCLK clock will also have the same HFCLK source. This means that parts of the core that have not requested the HFXO may get a clock that is more accurate than expected, but not the other way around.

All cores that have requested a HFCLK source to start by triggering a START task must also request it to stop by triggering the corresponding STOP task (see HFCLKSTOP, HFCLK192MSTOP, and HFCLKAUDIOSTOP tasks) before the power and clock subsystem will stop it.

HFCLK source(s) will stop when all corresponding STOP tasks have been triggered and there are no requests for HFCLK clock(s) from the system.

Triggering a HFCLK STOP task is required only if the corresponding HFCLK START task has been triggered before. When a HFCLK START task is triggered, it is possible to trigger again the same HFCLK START task without triggering the corresponding HFCLK STOP task in between.

4.11.1.1 Application core frequency scaling

The application core clocks can be scaled from their respective HFCLK clocks.

The application core clock HCLK128M can at any time be scaled from the HFCLK128M clock using the HFCLKCTRL register.

The HCLK192M clock can be scaled from the HFCLK192M clock using the HFCLK192MCTRL register.

Note: Settings Div1 and Div2 in HFCLK192MCTRL register will result in increased power consumption.

The ACLK audio clock cannot be scaled from the HFCLKAUDIO clock. Instead, its frequency can be configured in the relevant peripherals. Refer to Audio oscillator on page 72 for more information on audio clock and related peripherals.

Note: It is possible to scale the application core clocks at any time, for instance when a clock has already has been started, without having to stop it first.

4.11.1.2 32 MHz crystal oscillator (HFXO)

The 32 MHz crystal oscillator (HFXO) is controlled by a 32 MHz external crystal, see OSCILLATORS — Oscillator control on page 94.

4.11.1.3 Audio oscillator

The audio oscillator generates clock frequencies suitable for audio applications.

The audio oscillator has the following features:

- Adjustable frequency with 3.3 ppm resolution in two frequency bands - 11.176 MHz to 11.402 MHz, and 12.165 MHz to 12.411 MHz
- Low jitter, suitable for audio applications
- Always uses the HFXO

The HFCLKAUDIO clock generated by the audio oscillator is suitable for use as the source clock in the I^2S and PDM audio peripherals. In order to use this clock, it must be selected in the corresponding configuration registers in these peripherals. It is required to trigger the HFCLKAUDIOSTART task before it is used. To stop the HFCLKAUDIO clock, the HFCLKAUDIOSTOP task must be triggered. After triggering this task, the oscillator will be kept running as long as a peripheral is using it.

In applications where the audio data is arriving asynchronously to on-chip clocks, the frequency can be adjusted to stay in sync with the sender. The frequency can be configured in register HFCLKAUDIO.FREQUENCY on page 91 using one of the following equations.

$$f_{out} = \frac{32M}{12}(4 + HFCLKAUDIO.FREQUENCY \cdot 2^{-16})$$

Figure 19: Calculating audio frequency f_{out} from register value

$$HFCLKAUDIO.FREQUENCY = 2^{16}(\frac{12 \cdot f_{out}}{32M} - 4)$$

Figure 20: Calculating register value from audio frequency f_{out}

The acceptable HFCLKAUDIO.FREQUENCY register value ranges for the two frequency bands are listed in the following table.

When switching between the two frequency ranges, the peripherals must be stopped.

Frequency band	Register value and frequency		
	Min	Center	Max
11.176 MHz to 11.402 MHz	12519 (11.176 MHz)	15298 (11.289 MHz)	16068 (11.402 MHz)
12.165 MHz to 12.411 MHz	36834 (12.165 MHz)	39854 (12.288 MHz)	42874 (12.411 MHz)

Table 23: HFCLKAUDIO.FREQUENCY register ranges

4.11.1.4 Overriding the automatic HFCLK control system

Overriding the automatic clock control system is possible to ensure a HFCLK clock is started and kept running, even if not requested.

This can be used to avoid associated HFCLK clock start-up times and have the highest clock accuracy after wake-up from sleep.

The register HFCLKALWAYSRUN on page 91 can override the automatic clock control system for the HFCLK128M/HFCLK64M clocks. This override is initiated by performing the following steps:

1. Set HFCLKSRC.SRC to select the HFCLK source.
2. Set HFCLKALWAYSRUN.ALWAYSRUN.
3. Trigger the HFCLKSTART task.

The register HFCLK192MALWAYSRUN on page 92 can override the automatic clock control system for the HFCLK192M clock. This override is initiated by performing the following steps:

1. Set HFCLK192MSRC.SRC to select the HFCLK source.
2. Set HFCLK192MALWAYSRUN.ALWAYSRUN.
3. Trigger the HFCLK192MSTART task.

Registers HFCLKSRC/HFCLK192MSRC and HFCLKALWAYSRUN/HFCLK192MALWAYSRUN can be written at any time, but are only activated by the START task.

The register HFCLKAUDIOALWAYSRUN on page 92 can override the automatic clock control system for the HFCLKAUDIO clock. The override is initiated by performing the following steps:

1. Set HFCLKAUDIOALWAYSRUN.ALWAYSRUN.
2. Trigger the HFCLKAUDIOSTART task.

Note: In this case, the HFCLK source is always the HFXO.

Register HFCLKAUDIOALWAYSRUN can be written at any time, but is only activated by the START task.

4.11.2 LFCLK controller

Each core has a number of low frequency clock (LFCLK) control instances. Each instance distributes one or more clocks to the core.

The LFCLK control instance in each core distributes the 32.768 kHz PCLK32KI peripheral clock to its corresponding core. The LFCLK clock is sourced from the power and clock subsystem to each LFCLK control instance.

In order to generate the LFCLK clock, the LFCLK controller uses the following LFCLK sources:

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

For an illustration of the clock sources, see Clock control on page 69.

The LFCLK controller and all LFCLK sources are switched off in System OFF mode.

When peripherals require the PCLK32KI clock, such as RTC — Real-time counter on page 503 and WDT — Watchdog timer on page 737, the LFCLK control will automatically request the LFCLK clock to the power and clock subsystem. The default LFCLK source is the LFRC.

When LFCLK control requests are stopped, LFCLK will stop requesting clock from the power and clock subsystem. If there are no LFCLK control requests from other cores, the power and clock subsystem will automatically stop the LFCLK clock and the LFRC source.

The LFCLK source may also be started by triggering the LFCLKSTART task. The LFCLK source is configured by selecting the preferred LFCLK source in register LFCLKSRC on page 90. Once selected, the LFCLK source will be started by triggering the LFCLKSTART task.

The LFCLK source can be configured at any time (for instance, when the LFCLK has already been started). The content of the LFCLKSRC register only takes effect when the LFCLKSTART task is triggered.

Note: Automatic requests of the LFCLK clock will ignore the value in LFCLKSRC and use LFRC as source, unless the LFCLK source is started by triggering the LFCLKSTART start. In this case, the LFCLK source will correspond to the value in LFCLKSRC when the LFCLKSTART start was last triggered.

The LFCLKSTARTED event will be generated after the LFCLKSTART task has been triggered and the LFCLK source has started. Triggering a LFCLKSTART task before the LFCLKSTARTED event from a previous LFCLKSTART task is generated will only generate one LFCLKSTARTED event. Triggering a LFCLKSTART task after a LFCLKSTARTED event is generated will generate a new LFCLKSTARTED event.

The LFCLK clock is stopped when nothing requests it, e.g. RTC — Real-time counter on page 503 and WDT — Watchdog timer on page 737 are stopped, and the LFCLKSTOP task is triggered. This must be done for all cores. Triggering the LFCLKSTOP task is required only if the LFCLKSTART task has been triggered before.

When the LFCLKSTART task is triggered, it is possible to trigger a new LFCLKSTART task without triggering a LFCLKSTOP task in between.

If the LFXO is selected as the LFCLK source, the LFCLK clock will initially start running from the LFRC while the LFXO is starting up, and then automatically switch to using the LFXO once this oscillator is running.

Events will be generated in the correct order, even if an LFCLK source that is already started by another LFCLK control instance is requested. The timing of events may differ, depending on whether a LFCLK source is already running or in the process of starting.

If two instances of the LFCLK control system request different LFCLK sources, the power and clock subsystem will secure that the most accurate of the requested LFCLK sources is selected.

If one LFCLK control instance requests a particular LFCLK source to stop when another LFCLK control instance (or a peripheral) requests the same source to run, but at a lower accuracy, the power and clock subsystem will switch to the less accurate source. The following table summarizes the priorities of the LFCLK sources.

Priority	LFCLK source
Highest	LFSYNT
Second highest	LFXO
Lowest	LFRC

Table 24: LFCLK request priority

When switching the LFCLK source, such as from LFRC to LFXO, up to one LFCLK cycle may be lost.

4.11.2.1 32.768 kHz RC oscillator (LFRC)

An internal 32.768 kHz RC oscillator (LFRC) is available as the LFCLK source.

The LFRC oscillator is fully embedded in nRF5340 and does not require additional external components.

4.11.2.1.1 Calibrating the 32.768 kHz RC oscillator

To improve accuracy of the LFRC oscillator, it can be calibrated using the HFXO as a reference oscillator.

The LFRC oscillator can be calibrated while it is running. The calibration is started by triggering the CAL task which temporarily requests the HFCLK with the HFXO as the source for calibration.

A DONE event will be generated when the calibration is finished.

Note: Any core changing the LFCLK source will abort calibration without the DONE event being generated in the core triggering the CAL task.

If the CAL task is triggered while a calibration routine is already running (i.e. before the DONE event is generated), the CAL task has no effect and the calibration continues.

All cores can trigger the CAL task independently of each other. As a result, each core will receive a corresponding DONE event. If the calibration routine is already running (i.e. a core has triggered the CAL task), and the CAL task is triggered from another core, a DONE event is generated in both cores when the calibration of its corresponding LFRC oscillator is complete.

4.11.2.2 32.768 kHz crystal oscillator (LFXO)

For higher LFCLK accuracy (when greater than ± 250 ppm accuracy is required), the low frequency crystal oscillator (LFXO) must be used.

This clock source requires external components and GPIO pin configuration, see OSCILLATORS — Oscillator control on page 94.

4.11.2.3 32.768 kHz synthesized from HFCLK (LFSYNT)

The LFCLK can be synthesized from the HFCLK clock source.

LFSYNTH depends on the HFCLK to run. The accuracy of the LFCLK clock with the LFSYNTH as a source assumes the accuracy of the HFCLK. If high accuracy is required, the HFCLK must be generated from the HFXO.

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

4.11.2.4 Overriding the automatic LFCLK control system

Overriding the automatic clock control system to ensure the LFCLK clock is started and kept running is possible, even if not requested.

This can be used to avoid associated LFCLK clock start-up times and have the highest clock accuracy after wake-up from sleep.

The register LFCLKALWAYSRUN on page 91 can override the automatic clock control system. This override is initiated by performing the following steps:

1. Set LFCLKSRC.SRC to select the LFCLK source.
2. Set LFCLKALWAYSRUN.ALWAYSRUN.
3. Trigger the LFCLKSTART task.

Registers LFCLKSRC.SRC and LFCLKALWAYSRUN.ALWAYSRUN can be written at any time, but are only activated by the LFCLKSTART task.

4.11.3 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50005000 0x40005000	APPLICATION	CLOCK	CLOCK : S CLOCK : NS	US	NA	Clock control	
0x41005000	NETWORK	CLOCK	CLOCK	NS	NA	Clock control	HFCLKAUDIO not supported HFCLK192M not supported HFCLKCTRL reset value is 0x0.

Table 25: Instances

Register	Offset	Security	Description
TASKS_HFCLKSTART	0x000		Start HFCLK128M/HFCLK64M source as selected in HFCLKSRC
TASKS_HFCLKSTOP	0x004		Stop HFCLK128M/HFCLK64M source
TASKS_LFCLKSTART	0x008		Start LFCLK source as selected in LFCLKSRC
TASKS_LFCLKSTOP	0x00C		Stop LFCLK source
TASKS_CAL	0x010		Start calibration of LFRC oscillator
TASKS_HFCLKAUDIOSTART	0x018		Start HFCLKAUDIO source
TASKS_HFCLKAUDIOSTOP	0x01C		Stop HFCLKAUDIO source
TASKS_HFCLK192MSTART	0x020		Start HFCLK192M source as selected in HFCLK192MSRC
TASKS_HFCLK192MSTOP	0x024		Stop HFCLK192M source
SUBSCRIBE_HFCLKSTART	0x080		Subscribe configuration for task HFCLKSTART
SUBSCRIBE_HFCLKSTOP	0x084		Subscribe configuration for task HFCLKSTOP
SUBSCRIBE_LFCLKSTART	0x088		Subscribe configuration for task LFCLKSTART
SUBSCRIBE_LFCLKSTOP	0x08C		Subscribe configuration for task LFCLKSTOP
SUBSCRIBE_CAL	0x090		Subscribe configuration for task CAL
SUBSCRIBE_HFCLKAUDIOSTART	0x098		Subscribe configuration for task HFCLKAUDIOSTART
SUBSCRIBE_HFCLKAUDIOSTOP	0x09C		Subscribe configuration for task HFCLKAUDIOSTOP
SUBSCRIBE_HFCLK192MSTART	0x0A0		Subscribe configuration for task HFCLK192MSTART
SUBSCRIBE_HFCLK192MSTOP	0x0A4		Subscribe configuration for task HFCLK192MSTOP
EVENTS_HFCLKSTARTED	0x100		HFCLK128M/HFCLK64M source started
EVENTS_LFCLKSTARTED	0x104		LFCLK source started
EVENTS_DONE	0x11C		Calibration of LFRC oscillator complete event
EVENTS_HFCLKAUDIOSTARTED	0x120		HFCLKAUDIO source started
EVENTS_HFCLK192MSTARTED	0x124		HFCLK192M source started
PUBLISH_HFCLKSTARTED	0x180		Publish configuration for event HFCLKSTARTED
PUBLISH_LFCLKSTARTED	0x184		Publish configuration for event LFCLKSTARTED
PUBLISH_DONE	0x19C		Publish configuration for event DONE
PUBLISH_HFCLKAUDIOSTARTED	0x1A0		Publish configuration for event HFCLKAUDIOSTARTED
PUBLISH_HFCLK192MSTARTED	0x1A4		Publish configuration for event HFCLK192MSTARTED
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
INTPEND	0x30C		Pending interrupts

Register	Offset	Security	Description
HFCLKRUN	0x408		Status indicating that HFCLKSTART task has been triggered
HFCLKSTAT	0x40C		Status indicating which HFCLK128M/HFCLK64M source is running This register value in any CLOCK instance reflects status only due to configurations/ actions in that CLOCK instance.
LFCLKRUN	0x414		Status indicating that LFCLKSTART task has been triggered
LFCLKSTAT	0x418		Status indicating which LFCLK source is running This register value in any CLOCK instance reflects status only due to configurations/ actions in that CLOCK instance.
LFCLKSRCCOPY	0x41C		Copy of LFCLKSRC register, set when LFCLKSTART task was triggered
HFCLKAUDIORUN	0x450		Status indicating that HFCLKAUDIOSTART task has been triggered
HFCLKAUDIOSTAT	0x454		Status indicating which HFCLKAUDIO source is running
HFCLK192MRUN	0x458		Status indicating that HFCLK192MSTART task has been triggered
HFCLK192MSTAT	0x45C		Status indicating which HFCLK192M source is running
HFCLKSRC	0x514		Clock source for HFCLK128M/HFCLK64M
LFCLKSRC	0x518		Clock source for LFCLK
HFCLKCTRL	0x558		HFCLK128M frequency configuration
HFCLKAUDIO.FREQUENCY	0x55C		Audio PLL frequency in 11.176 MHz - 11.402 MHz or 12.165 MHz - 12.411 MHz frequency bands
HFCLKALWAYSRUN	0x570		Automatic or manual control of HFCLK128M/HFCLK64M
LFCLKALWAYSRUN	0x574		Automatic or manual control of LFCLK
HFCLKAUDIOALWAYSRUN	0x57C		Automatic or manual control of HFCLKAUDIO
HFCLK192MSRC	0x580		Clock source for HFCLK192M
HFCLK192MALWAYSRUN	0x584		Automatic or manual control of HFCLK192M
HFCLK192MCTRL	0x5B8		HFCLK192M frequency configuration

Table 26: Register overview

4.11.3.1 TASKS_HFCLKSTART

Address offset: 0x000

Start HFCLK128M/HFCLK64M source as selected in HFCLKSRC

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLKSTART			Start HFCLK128M/HFCLK64M source as selected in HFCLKSRC
			Trigger	1	Trigger task

4.11.3.2 TASKS_HFCLKSTOP

Address offset: 0x004

Stop HFCLK128M/HFCLK64M 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	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLKSTOP			Stop HFCLK128M/HFCLK64M source
			Trigger	1	Trigger task

4.11.3.3 TASKS_LFCLKSTART

Address offset: 0x008

Start LFCLK source as selected in LFCLKSRC

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_LFCLKSTART			Start LFCLK source as selected in LFCLKSRC
			Trigger	1	Trigger task

4.11.3.4 TASKS_LFCLKSTOP

Address offset: 0x00C

Stop LFCLK 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					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_LFCLKSTOP			Stop LFCLK source
			Trigger	1	Trigger task

4.11.3.5 TASKS_CAL

Address offset: 0x010

Start calibration of LFRC oscillator

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_CAL			Start calibration of LFRC oscillator
			Trigger	1	Trigger task

4.11.3.6 TASKS_HFCLKAUDIOSTART

Address offset: 0x018

Start HFCLKAUDIO 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					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLKAUDIOSTART			Start HFCLKAUDIO source
			Trigger	1	Trigger task

4.11.3.7 TASKS_HFCLKAUDIOSTOP

Address offset: 0x01C

Stop HFCLKAUDIO 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLKAUDIOSTOP			Stop HFCLKAUDIO source
			Trigger	1	Trigger task

4.11.3.8 TASKS_HFCLK192MSTART

Address offset: 0x020

Start HFCLK192M source as selected in HFCLK192MSRC

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLK192MSTART			Start HFCLK192M source as selected in HFCLK192MSRC
			Trigger	1	Trigger task

4.11.3.9 TASKS_HFCLK192MSTOP

Address offset: 0x024

Stop HFCLK192M 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_HFCLK192MSTOP			Stop HFCLK192M source
			Trigger	1	Trigger task

4.11.3.10 SUBSCRIBE_HFCLKSTART

Address offset: 0x080

Subscribe configuration for task HFCLKSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLKSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.11 SUBSCRIBE_HFCLKSTOP

Address offset: 0x084

Subscribe configuration for task HFCLKSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLKSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.12 SUBSCRIBE_LFCLKSTART

Address offset: 0x088

Subscribe configuration for task LFCLKSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task LFCLKSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.13 SUBSCRIBE_LFCLKSTOP

Address offset: 0x08C

Subscribe configuration for task LFCLKSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task LFCLKSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.14 SUBSCRIBE_CAL

Address offset: 0x090

Subscribe configuration for task CAL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CAL will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.15 SUBSCRIBE_HFCLKAUDIOSTART

Address offset: 0x098

Subscribe configuration for task HFCLKAUDIOSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLKAUDIOSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.16 SUBSCRIBE_HFCLKAUDIOSTOP

Address offset: 0x09C

Subscribe configuration for task HFCLKAUDIOSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLKAUDIOSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.17 SUBSCRIBE_HFCLK192MSTART

Address offset: 0x0A0

Subscribe configuration for task HFCLK192MSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLK192MSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.18 SUBSCRIBE_HFCLK192MSTOP

Address offset: 0x0A4

Subscribe configuration for task HFCLK192MSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task HFCLK192MSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.11.3.19 EVENTS_HFCLKSTARTED

Address offset: 0x100

HFCLK128M/HFCLK64M source started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_HFCLKSTARTED			HFCLK128M/HFCLK64M source started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.11.3.20 EVENTS_LFCLKSTARTED

Address offset: 0x104

LFCLK source started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_LFCLKSTARTED			LFCLK source started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.11.3.21 EVENTS_DONE

Address offset: 0x11C

Calibration of LFRC oscillator complete 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_DONE			Calibration of LFRC oscillator complete event
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.11.3.22 EVENTS_HFCLKAUDIOSTARTED

Address offset: 0x120

HFCLKAUDIO source started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_HFCLKAUDIOSTARTED			HFCLKAUDIO source started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.11.3.23 EVENTS_HFCLK192MSTARTED

Address offset: 0x124

HFCLK192M source started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_HFCLK192MSTARTED			HFCLK192M source started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.11.3.24 PUBLISH_HFCLKSTARTED

Address offset: 0x180

Publish configuration for event HFCLKSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event HFCLKSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.11.3.25 PUBLISH_LFCLKSTARTED

Address offset: 0x184

Publish configuration for event LFCLKSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event LFCLKSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.11.3.26 PUBLISH_DONE

Address offset: 0x19C

Publish configuration for event DONE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.11.3.27 PUBLISH_HFCLKAUDIOSTARTED

Address offset: 0x1A0

Publish configuration for event HFCLKAUDIOSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event HFCLKAUDIOSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.11.3.28 PUBLISH_HFCLK192MSTARTED

Address offset: 0x1A4

Publish configuration for event HFCLK192MSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event HFCLK192MSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.11.3.29 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				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	HFCLKSTARTED			Enable or disable interrupt for event HFCLKSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	LFCLKSTARTED			Enable or disable interrupt for event LFCLKSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	DONE			Enable or disable interrupt for event DONE
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	HFCLKAUDIOSTARTED			Enable or disable interrupt for event HFCLKAUDIOSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	HFCLK192MSTARTED			Enable or disable interrupt for event HFCLK192MSTARTED
			Disabled	0	Disable
			Enabled	1	Enable

4.11.3.30 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	HFCLKSTARTED			Write '1' to enable interrupt for event HFCLKSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	LFCLKSTARTED			Write '1' to enable interrupt for event LFCLKSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to enable interrupt for event DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	HFCLKAUDIOSTARTED			Write '1' to enable interrupt for event HFCLKAUDIOSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	HFCLK192MSTARTED			Write '1' to enable interrupt for event HFCLK192MSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.11.3.31 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	HFCLKSTARTED			Write '1' to disable interrupt for event HFCLKSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	LFCLKSTARTED			Write '1' to disable interrupt for event LFCLKSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to disable interrupt for event DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	HFCLKAUDIOSTARTED			Write '1' to disable interrupt for event HFCLKAUDIOSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	HFCLK192MSTARTED			Write '1' to disable interrupt for event HFCLK192MSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.11.3.32 INTPEND

Address offset: 0x30C

Pending interrupts

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

ID	R/W	Field	Value ID	Value	Description
A	R	HFCLKSTARTED			Read pending status of interrupt for event HFCLKSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B	R	LFCLKSTARTED			Read pending status of interrupt for event LFCLKSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
C	R	DONE			Read pending status of interrupt for event DONE
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
D	R	HFCLKAUDIOSTARTED			Read pending status of interrupt for event HFCLKAUDIOSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
E	R	HFCLK192MSTARTED			Read pending status of interrupt for event HFCLK192MSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

4.11.3.33 HFCLKRUN

Address offset: 0x408

Status indicating that HFCLKSTART task has been triggered

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

4.11.3.34 HFCLKSTAT

Address offset: 0x40C

Status indicating which HFCLK128M/HFCLK64M source is running

This register value in any CLOCK instance reflects status only due to configurations/actions in that CLOCK instance.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	SRC			Active clock source
			HFINT	0	Clock source: HFINT - 128 MHz on-chip oscillator
			HFXO	1	Clock source: HFXO - 128 MHz clock derived from external 32 MHz crystal oscillator
B	R	ALWAYSRUNNING			ALWAYSRUN activated
			NotRunning	0	Automatic clock control enabled
			Running	1	Oscillator is always running
C	R	STATE			HFCLK state
			NotRunning	0	HFCLK not running
			Running	1	HFCLK running

4.11.3.35 LFCLKRUN

Address offset: 0x414

Status indicating that LFCLKSTART task has been triggered

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

ID	R/W	Field	Value ID	Value	Description
A	R	STATUS			LFCLKSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

4.11.3.36 LFCLKSTAT

Address offset: 0x418

Status indicating which LFCLK source is running

This register value in any CLOCK instance reflects status only due to configurations/actions in that CLOCK instance.

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

ID	R/W	Field	Value ID	Value	Description
A	R	SRC			Active clock source
			LFRC	1	32.768 kHz RC oscillator
			LFXO	2	32.768 kHz crystal oscillator
			LFSYNT	3	32.768 kHz synthesized from HFCLK
B	R	ALWAYSRUNNING			ALWAYSRUN activated
			NotRunning	0	Automatic clock control enabled
			Running	1	Oscillator is always running
C	R	STATE			LFCLK state
			NotRunning	0	LFCLK not running
			Running	1	LFCLK running

4.11.3.37 LFCLKSRCCOPY

Address offset: 0x41C

Copy of LFCLKSRC register, set when LFCLKSTART task was triggered

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

ID	R/W	Field	Value ID	Value	Description
A	R	SRC			Clock source
			LFRC	1	32.768 kHz RC oscillator
			LFXO	2	32.768 kHz crystal oscillator
			LFSYNT	3	32.768 kHz synthesized from HFCLK

4.11.3.38 HFCLKAUDIORUN

Address offset: 0x450

Status indicating that HFCLKAUDIOSTART task has been triggered

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

ID	R/W	Field	Value ID	Value	Description
A	R	STATUS			HFCLKAUDIOSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

4.11.3.39 HFCLKAUDIOSTAT

Address offset: 0x454

Status indicating which HFCLKAUDIO source is running

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

ID	R/W	Field	Value ID	Value	Description
B	R	ALWAYSRUNNING			ALWAYSRUN activated
			NotRunning	0	Automatic clock control enabled
			Running	1	Oscillator is always running
C	R	STATE			HFCLKAUDIO state
			NotRunning	0	HFCLKAUDIO not running
			Running	1	HFCLKAUDIO running

4.11.3.40 HFCLK192MRUN

Address offset: 0x458

Status indicating that HFCLK192MSTART task has been triggered

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

ID	R/W	Field	Value ID	Value	Description
A	R	STATUS			HFCLK192MSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

4.11.3.41 HFCLK192MSTAT

Address offset: 0x45C

Status indicating which HFCLK192M source is running

Bit number				31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A	R	SRC			Active clock source
			HFINT	0	Clock source: HFINT - on-chip oscillator
			HFXO	1	Clock source: HFXO - derived from external 32 MHz crystal oscillator
B	R	ALWAYSRUNNING			ALWAYSRUN activated
			NotRunning	0	Automatic clock control enabled
			Running	1	Oscillator is always running
C	R	STATE			HFCLK192M state
			NotRunning	0	HFCLK192M not running
			Running	1	HFCLK192M running

4.11.3.42 HFCLKSRC

Address offset: 0x514

Clock source for HFCLK128M/HFCLK64M

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SRC			Select which HFCLK source is started by the HFCLKSTART task
			HFINT	0	HFCLKSTART task starts HFINT oscillator
			HFXO	1	HFCLKSTART task starts HFXO oscillator

4.11.3.43 LFCLKSRC

Address offset: 0x518

Clock source for LFCLK

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SRC			Select which LFCLK source is started by the LFCLKSTART task
			LFRC	1	32.768 kHz RC oscillator
			LFXO	2	32.768 kHz crystal oscillator
			LFSYNT	3	32.768 kHz synthesized from HFCLK

4.11.3.44 HFCLKCTRL

Address offset: 0x558

HFCLK128M frequency configuration

Using any value except for the enumerations will yield unexpected results

Note: Not present in the CLOCK instance of the network core

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	HCLK			High frequency clock HCLK
			Div1	0	Divide HFCLK by 1
			Div2	1	Divide HFCLK by 2

4.11.3.45 HFCLKAUDIO.FREQUENCY

Address offset: 0x55C

Audio PLL frequency in 11.176 MHz - 11.402 MHz or 12.165 MHz - 12.411 MHz frequency bands

Note: Not present in the CLOCK instance of the network core

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00009BAE				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 1 0 1 1 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	FREQUENCY			Frequency
					0: 10.666 MHz
					65535: 13.333 MHz

4.11.3.46 HFCLKALWAYSRUN

Address offset: 0x570

Automatic or manual control of HFCLK128M/HFCLK64M

The AlwaysRun setting will ensure the clock source is always running, independent of the automatic clock request system.

Note: This setting is activated by triggering the HFCLKSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ALWAYSRUN			Ensure clock is always running
			Automatic	0	Use automatic clock control
			AlwaysRun	1	Ensure clock is always running

4.11.3.47 LFCLKALWAYSRUN

Address offset: 0x574

Automatic or manual control of LFCLK

The AlwaysRun setting will ensure the clock source is always running, independent of the automatic clock request system.

Note: This setting is activated by triggering the LFCLKSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ALWAYSRUN			Ensure clock is always running
			Automatic	0	Use automatic clock control
			AlwaysRun	1	Ensure clock is always running

4.11.3.48 HFCLKAUDIOALWAYSRUN

Address offset: 0x57C

Automatic or manual control of HFCLKAUDIO

The AlwaysRun setting will ensure the clock source is always running, independent of the automatic clock request system.

Note: This setting is activated by triggering the HFCLKAUDIOSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ALWAYSRUN			Ensure clock is always running
			Automatic	0	Use automatic clock control
			AlwaysRun	1	Ensure clock is always running

4.11.3.49 HFCLK192MSRC

Address offset: 0x580

Clock source for HFCLK192M

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SRC			Select which HFCLK192M source is started by the HFCLK192MSTART task
			HFINT	0	HFCLK192MSTART task starts HFINT oscillator
			HFXO	1	HFCLK192MSTART task starts HFXO oscillator

4.11.3.50 HFCLK192MALWAYSRUN

Address offset: 0x584

Automatic or manual control of HFCLK192M

The AlwaysRun setting will ensure the clock source is always running, independent of the automatic clock request system.

Note: This setting is activated by triggering the HFCLK192MSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ALWAYSRUN			Ensure clock is always running
			Automatic	0	Use automatic clock control
			AlwaysRun	1	Ensure clock is always running

4.11.3.51 HFCLK192MCTRL

Address offset: 0x5B8

HFCLK192M frequency 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 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	HCLK192M			High frequency clock HCLK192M
			Div1	0	Divide HFCLK192M by 1
			Div2	1	Divide HFCLK192M by 2
			Div4	2	Divide HFCLK192M by 4

4.11.4 Electrical specification

4.11.4.1 128 MHz clock source (HFCLK128M)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{NOM\_HFCLK128M}$	Nominal output frequency		128		MHz
$f_{TOL\_HFINT128M}$	Frequency tolerance when running from internal oscillator		±1.5	±7	%
$f_{TOL\_HFXO128M}$	Frequency tolerance when running from crystal oscillator		±10	±60	ppm

4.11.4.2 64 MHz clock source (HFCLK64M)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{NOM\_HFCLK64M}$	Nominal output frequency		64		MHz
$f_{TOL\_HFINT64M}$	Frequency tolerance when running from internal oscillator		±1.5	±8	%
$f_{TOL\_HFXO64M}$	Frequency tolerance when running from crystal oscillator		±10	±60	ppm

4.11.4.3 192 MHz clock source (HFCLK192M)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{NOM\_HFCLK192M}$	Nominal output frequency		192		MHz
$f_{TOL\_HFINT192M}$	Frequency tolerance when running from internal oscillator		±1.5	±7	%
$f_{TOL\_HFXO192M}$	Frequency tolerance when running from crystal oscillator		±10	±60	ppm

4.11.4.4 Audio clock source (HFCLKAUDIO)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{NOM\_HFCLKAUDIO}$	Nominal output frequency		11.289 or 12.288		MHz
$f_{TOL\_HFXOAUDIO}$	Frequency tolerance when running from crystal oscillator		±10	±60	ppm

4.11.4.5 32 kHz clock source (LFCLK)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_LFCLK}	Nominal output frequency		32.768		kHz
t_{START_LFXO}	Startup time for 32.768 kHz crystal oscillator		0.31		s
I_{LFXO}	Run current for 32.768 kHz crystal oscillator		0.16		µA
f_{TOL_LFRC}	Frequency tolerance, uncalibrated			±3.2	%
$f_{TOL\_CAL\_LFRC}$	Frequency tolerance after calibration. Constant temperature within ±0.5 °C, calibration performed at least every 8 seconds, averaging interval > 7.5 ms, defined as 3 sigma.			±250	ppm
t_{START_LFRC}	Startup time for internal RC oscillator		500		µs
I_{LFRC}	Run current for LFRC			1.0	µA

4.12 OSCILLATORS — Oscillator control

The system oscillators are shared between the cores in the system and automatically controlled by the clock control system, see CLOCK — Clock control on page 69.

The system has the following crystal oscillators:

- High-frequency 32 MHz crystal oscillator (HFXO)
- Low-frequency 32.768 kHz crystal oscillator (LFXO)

The crystal oscillators can be configured to use either built-in or external capacitors.

4.12.1 High-frequency (32 MHz) crystal oscillator (HFXO)

The high-frequency 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, connected between pins XC1 and XC2. To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet. The following figure shows how the 32 MHz crystal is connected to the high frequency crystal oscillator.

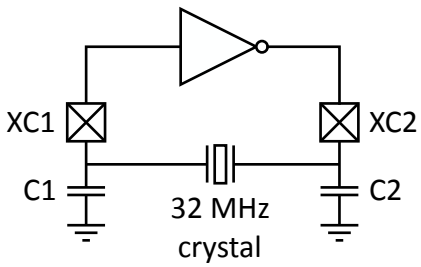

Figure 21: Circuit diagram of the high-frequency crystal oscillator

The load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is calculated by the following equation.

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

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

Figure 22: Load capacitance equation

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. For more information, see Reference circuitry on page 788. C_{pcb1} and C_{pcb2} are stray capacitances on the PCB. C_{pin} is the pin input capacitance on pins XC1 and XC2. See table High-frequency (32 MHz) crystal oscillator (HFXO) on page 94. 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 High-frequency (32 MHz) crystal oscillator (HFXO) on page 94. It is recommended to use a crystal with lower than maximum load capacitance and/or shunt capacitance. A low load capacitance will reduce both startup time and current consumption.

4.12.1.1 Using internal capacitors

Optional internal capacitors ranging from 7.0 pF to 20.0 pF in 0.5 pF steps, are provided on pins XC1 and XC2.

Enabling internal capacitors eliminates the need for external capacitors for the 32 MHz crystal. The configuration of the internal capacitors must take place before starting the high-frequency crystal oscillator (HFXO).

The internal capacitors are used instead of the external capacitors C1 and C2, and the total capacitance seen by the crystal across its terminals is calculated by the load capacitance equation in High-frequency (32 MHz) crystal oscillator (HFXO) on page 94.

To enable the internal capacitors, find the correct XOSC32MCAPS.CAPVALUE field using the following equation.

```
CAPVALUE = (((FICR->XOSC32MTRIM.SLOPE+56)*(CAPACITANCE*2-14))
           +((FICR->XOSC32MTRIM.OFFSET-8)<<4)+32)>>6;
```

The equation has the following variables

- `CAPACITANCE` is the desired capacitor value in pF, holding any value between 7.0 pF and 20.0 pF in 0.5 pF steps.
- `FICR->XOSC32MTRIM` are factory trim values which usually are different from device to device.

Finally, set XOSC32MCAPS.ENABLE to `Enabled`, to activate the capacitors.

After this, when HFXO is started, it will use the internal capacitor values together with the external crystal.

Note: It is possible to avoid using floating point numbers by pre-calculating the `(CAPACITANCE*2-14)` field of the above equation.

4.12.2 Low-frequency (32.768 kHz) crystal oscillator (LFXO)

For higher LFCLK accuracy (when greater than ± 250 ppm accuracy is required), the 32.768 kHz crystal oscillator (LFXO) must be used.

To use the LFXO, a 32.768 kHz crystal must be connected between the XL1 and XL2 pins, as illustrated in the following figure.

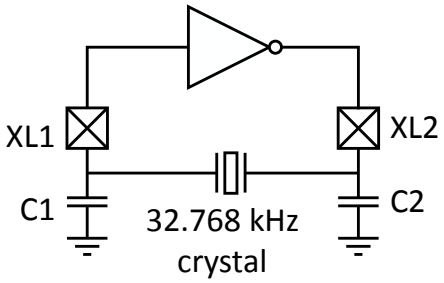

Figure 23: Circuit diagram of the low-frequency crystal oscillator

To enable oscillator functionality on XL1 and XL2 pins, use value *Peripheral* for the MCUSEL bitfield of the register PIN_CNF[n] (n=0..31) (Retained) on page 228.

To achieve correct oscillation frequency, the load capacitance (CL) must match the specification in the crystal data sheet. The load capacitance (CL) is the total capacitance seen by the crystal across its terminals. It is calculated by the following equation.

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

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

Figure 24: Load capacitance equation

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 XL1 and XL2 pins (see Low-frequency (32.768 kHz) crystal oscillator (LFXO) on page 95). The load capacitors C1 and C2 should have the same value.

For more information, see Reference circuitry on page 788.

4.12.2.1 Using internal capacitors

Optional internal capacitors of 6 pF, 7 pF, and 9 pF are provided on pins XL1 and XL2.

To use these capacitors, select the capacitance in register XOSC32KI.INTCAP.

4.12.3 Low-frequency (32.768 kHz) external source

The 32.768 kHz crystal oscillator (LFXO) is designed to work with external sources.

The following external sources are supported:

- A low swing clock. The signal should be applied to the XL1 pin with the XL2 pin grounded. Set OSCILLATORS.XOSC32KI.BYPASS=Disabled.
- A rail-to-rail clock. The signal should be applied to the XL1 pin with the XL2 pin left unconnected. Set OSCILLATORS.XOSC32KI.BYPASS=Enabled.

In order to use an external source, configure LFCLKSRC.SRC=LFXO.

4.12.4 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50004000 0x40004000	APPLICATION	OSCILLATORS	OSCILLATORS : S OSCILLATORS : NS	US	NA	Oscillator configuration	

Table 27: Instances

Register	Offset	Security	Description	
XOSC32MCAPS	0x5C4		Programmable capacitance of XC1 and XC2	Retained
XOSC32KI.BYPASS	0x6C0		Enable or disable bypass of LFCLK crystal oscillator with external clock source	Retained
XOSC32KI.INTCAP	0x6D0		Control usage of internal load capacitors	Retained

Table 28: Register overview

4.12.4.1 XOSC32MCAPS (Retained)

Address offset: 0x5C4

This register is a retained register

Programmable capacitance of XC1 and XC2

Bit number				31 30 29 28 27 26 25 24 23 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 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CAPVALUE			Value representing capacitance, calculated using provided equation
B	RW	ENABLE			Enable on-chip capacitors on XC1 and XC2
			Disabled	0	Capacitor disabled (use external caps)
			Enabled	1	Capacitor enabled

4.12.4.2 XOSC32KI.BYPASS (Retained)

Address offset: 0x6C0

This register is a retained register

Enable or disable bypass of LFCLK crystal oscillator with external clock 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	BYPASS			Enable or disable bypass of LFCLK crystal oscillator with external clock source
			Disabled	0	Disable (use with crystal or low-swing external source)
			Enabled	1	Enable (use with rail-to-rail external source)

4.12.4.3 XOSC32KI.INTCAP (Retained)

Address offset: 0x6D0

This register is a retained register

Control usage of internal load capacitors

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value																															
A	RW	INTCAP																																	
			External	0																															
			C6PF	1																															
			C7PF	2																															
			C9PF	3																															

4.12.5 Electrical specification

4.12.5.1 32 MHz crystal oscillator (HFXO)

Symbol	Description	Min.	Typ.	Max.	Units
f_{HFXO}	External crystal frequency		32		MHz
f_{TOL_HFXO}	Frequency tolerance requirement for 2.4 GHz proprietary radio applications			±60	ppm
$f_{TOL\_HFXO\_BLE}$	Frequency tolerance requirement, Bluetooth Low Energy applications, packet length ≤ 200 bytes			±40	ppm
$f_{TOL\_HFXO\_BLE\_LP}$	Frequency tolerance requirement, Bluetooth Low Energy applications, packet length > 200 bytes			±30	ppm
C_{L_HFXO}	Load capacitance	6	8	12	pF
C_{0_HFXO}	Shunt capacitance			7	pF
$R_{S\_HFXO\_7PF}$	Equivalent series resistance 3 pF < C0 <= 7 pF			60	Ω
$R_{S\_HFXO\_3PF}$	Equivalent series resistance C0 <= 3 pF			100	Ω
P_{D_HFXO}	Drive level			100	µW
C_{PIN_HFXO}	Input capacitance XC1 and XC2 with internal capacitors disabled		5.5		pF
$C_{PIN\_HFXO\_INT}$	Input capacitance XC1 and XC2 with internal capacitors enabled		2.5		pF
$C_{HFXO\_INT\_MIN}$	Input capacitance XC1 and XC2, internal capacitor at minimum value, excluding $C_{PIN\_HFXO\_INT}$		7		pF
$C_{HFXO\_INT\_MAX}$	Input capacitance XC1 and XC2, internal capacitor at maximum value, excluding $C_{PIN\_HFXO\_INT}$		20		pF
I_{STBY_X32M}	Core standby current for various crystals				
$I_{STBY\_X32M\_X0}$	Typical parameters for a given 2.5mm x 2.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 1 pF, LM_HFXO = 7 mH, RS_HFXO = 20 Ω		65		µA
$I_{STBY\_X32M\_X1}$	Typical parameters for a given 1.6mm x 1.2mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.4 pF, LM_HFXO = 20 mH, RS_HFXO = 40 Ω		187		µA
$I_{STBY\_X32M\_X2}$	Typical parameters for a given 2.0mm x 1.6mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.73 pF, LM_HFXO = 9.47 mH, RS_HFXO = 16.32 Ω		135		µA

Symbol	Description	Min.	Typ.	Max.	Units
$I_{STBY\_X32M\_X3}$	Typical parameters for a given 1.2mm x 1.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.42 pF, LM_HFXO = 22.7 mH, RS_HFXO = 100 Ω		181		µA
I_{START_X32M}	Average startup current for various crystals, first 1 ms				
$I_{START\_X32M\_X0}$	Typical parameters for a given 2.5mm x 2.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 1 pF, LM_HFXO = 7 mH, RS_HFXO = 20 Ω		363		µA
$I_{START\_X32M\_X1}$	Typical parameters for a given 1.6mm x 1.2mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.4 pF, LM_HFXO = 20 mH, RS_HFXO = 40 Ω		790		µA
$I_{START\_X32M\_X2}$	Typical parameters for a given 2.0mm x 1.6mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.73 pF, LM_HFXO = 9.47 mH, RS_HFXO = 16.32 Ω		467		µA
$I_{START\_X32M\_X3}$	Typical parameters for a given 1.2mm x 1.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.42 pF, LM_HFXO = 22.7 mH, RS_HFXO = 100 Ω		863		µA
$t_{POWERUP\_X32M}$	Power-up time for various crystals				
$t_{POWERUP\_X32M\_X0}$	Typical parameters for a given 2.5mm x 2.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 1 pF, LM_HFXO = 7 mH, RS_HFXO = 20 Ω		60		µs
$t_{POWERUP\_X32M\_X1}$	Typical parameters for a given 1.6mm x 1.2mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.4 pF, LM_HFXO = 20 mH, RS_HFXO = 40 Ω		187		µs
$t_{POWERUP\_X32M\_X2}$	Typical parameters for a given 2.0mm x 1.6mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.73 pF, LM_HFXO = 9.47 mH, RS_HFXO = 16.32 Ω		60		µs
$t_{POWERUP\_X32M\_X3}$	Typical parameters for a given 1.2mm x 1.0mm crystal: CL_HFXO = 8 pF, C0_HFXO = 0.42 pF, LM_HFXO = 22.7 mH, RS_HFXO = 100 Ω		211		µs

4.12.5.2 32.768 kHz crystal oscillator (LFXO)

Symbol	Description	Min.	Typ.	Max.	Units
f_{LFXO}	External crystal frequency		32.768		kHz
$f_{TOL\_LFXO\_BLE}$	Frequency tolerance requirement, Bluetooth Low Energy applications			±500	ppm
$f_{TOL\_LFXO\_ANT}$	Frequency tolerance requirement for ANT applications			±50	ppm
C_{L_LFXO}	Load capacitance		7	9	pF
C_{0_LFXO}	Shunt capacitance		1	2	pF
R_{S_LFXO}	Equivalent series resistance		60	90	kΩ
P_{D_LFXO}	Drive level			0.5	µW
C_{pin}	Input capacitance on XL1 and XL2 pads with internal capacitor disabled		4		pF
C_{pin}	Input capacitance on XL1 and XL2 pads with internal capacitor enabled	6	7	9	pF

5 Application core

The application core contains a low-power microcontroller with embedded flash memory and a full featured Arm Cortex-M33 processor.

In addition, the application core includes a rich set of peripherals for serial communication, analog interfaces, and cryptographic acceleration. See the following figure for more information. Arrows with white heads indicate signals that share physical pins with other signals, while arrows with black heads are dedicated to one signal.

5.1 Block diagram

The following image shows the application core block diagram.

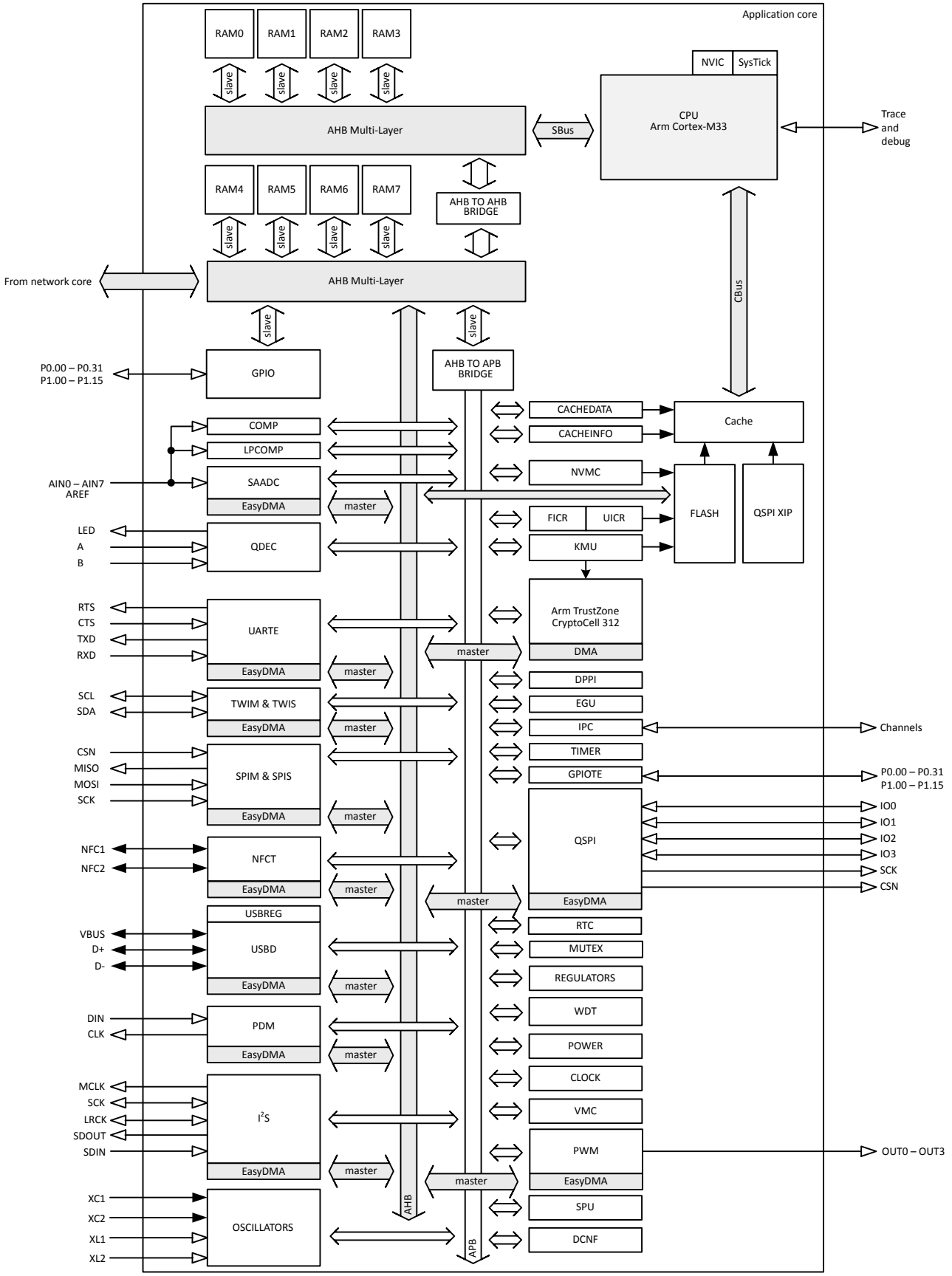

Figure 25: Application core block diagram

5.2 CPU

The Arm Cortex-M33 processor has a 32-bit instruction set (Thumb-2 technology) that implements a superset of 16- and 32-bit instructions to maximize code density and performance.

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

- Digital signal processing (DSP) instructions
- Single-cycle multiply and accumulate (MAC) instructions
- Hardware divide
- 8- and 16-bit single instruction multiple data (SIMD) instructions
- Single-precision floating-point unit (FPU)
- Memory Protection Unit (MPU)
- Arm TrustZone for Armv8-M

The Arm Cortex Microcontroller Software Interface Standard (CMSIS) is implemented and available for the application processor.

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 internal or external flash will have a wait state penalty. The instruction cache can be enabled to minimize flash wait states when fetching instructions. For more information on cache, see Cache on page 332. CPU performance parameters including wait states for different configurations, CPU current consumpion and efficiency, and processing power and efficiency based on the CoreMark benchmark can be found in Electrical specification on page 102.

5.2.1 Floating point interrupt

The floating point unit (FPU) may generate exceptions, for example, due to overflow or underflow. These exceptions may trigger interrupts when enabled in the FPU peripheral. For more information, see FPU - Floating point unit (FPU) exceptions on page 214.

5.2.2 Electrical specification

5.2.2.1 CPU performance

Symbol	Description	Min.	Typ.	Max.	Units
$W_{FLASH128}$	CPU wait states, running CoreMark at 128 MHz from flash, cache disabled			4	
$W_{FLASHCACHE128}$	CPU wait states, running CoreMark at 128 MHz from flash, cache enabled			5	
W_{RAM128}	CPU wait states, running CoreMark at 128 MHz from RAM		0		
$W_{FLASH64}$	CPU wait states, running CoreMark at 64 MHz from flash, cache disabled			5	
$W_{FLASHCACHE64}$	CPU wait states, running CoreMark at 64 MHz from flash, cache enabled			6	
W_{RAM64}	CPU wait states, running CoreMark at 64 MHz from RAM		0		
$CM_{FLASHCACHE128}$	CoreMark, running from flash, cache enabled, HFXO128M		514		CoreMark
$CM_{FLASH128/MHz}$	CoreMark per MHz, running from flash, cache enabled, HFXO128M		4.0		CoreMark/MHz
$CM_{FLASH128/mA}$	CoreMark per mA, running from flash, cache enabled, DCDC 3 V, HFXO128M		66		CoreMark/mA

Symbol	Description	Min.	Typ.	Max.	Units
$CM_{FLASHCACHE64}$	CoreMark, running from flash, cache enabled, HFXO64M		257		CoreMark
$CM_{FLASH64/MHz}$	CoreMark per MHz, running from flash, cache enabled, HFXO64M		4.0		CoreMark/MHz
$CM_{FLASH64/mA}$	CoreMark per mA, running from flash, cache enabled, DCDC 3 V, HFXO64M		72.5		CoreMark/mA

5.2.3 CPU and support module configuration

The Arm Cortex-M33 processor has a number of CPU options and support modules implemented on the device.

Option/Module	Description	Implemented
Core options		
PRIORITIES	Priority bits	3
WIC	Wakeup Interrupt Controller	NO
Endianness	Memory system endianness	Little endian
DWT	Data Watchpoint and Trace	YES
Modules		
MPU	Number of non-secure MPU regions	8
	Number of secure MPU regions	8
SAU	Number of SAU regions	0 See SPU for more information about secure regions.
FPU	Floating-point unit	YES
DSP	Digital Signal Processing Extension	YES
Arm TrustZone for Armv8-M	Armv8-M Security Extensions	YES
CPIF	Coprocessor interface	NO
ETM	Embedded Trace Macrocell	YES
ITM	Instrumentation Trace Macrocell	YES
MTB	Micro Trace Buffer	NO
CTI	Cross Trigger Interface	YES
BPU	Breakpoint Unit	YES
HTM	AHB Trace Macrocell	NO

5.3 Memory

The application core contains flash memory and RAM that can be used for code and data storage.

The following figure shows how the CPU, network core, and peripherals with EasyDMA can access RAM via the AHB multilayer interconnect. The domain configuration (DCNF) registers can block access from external DMA masters, see DCNF — Domain configuration on page 196.

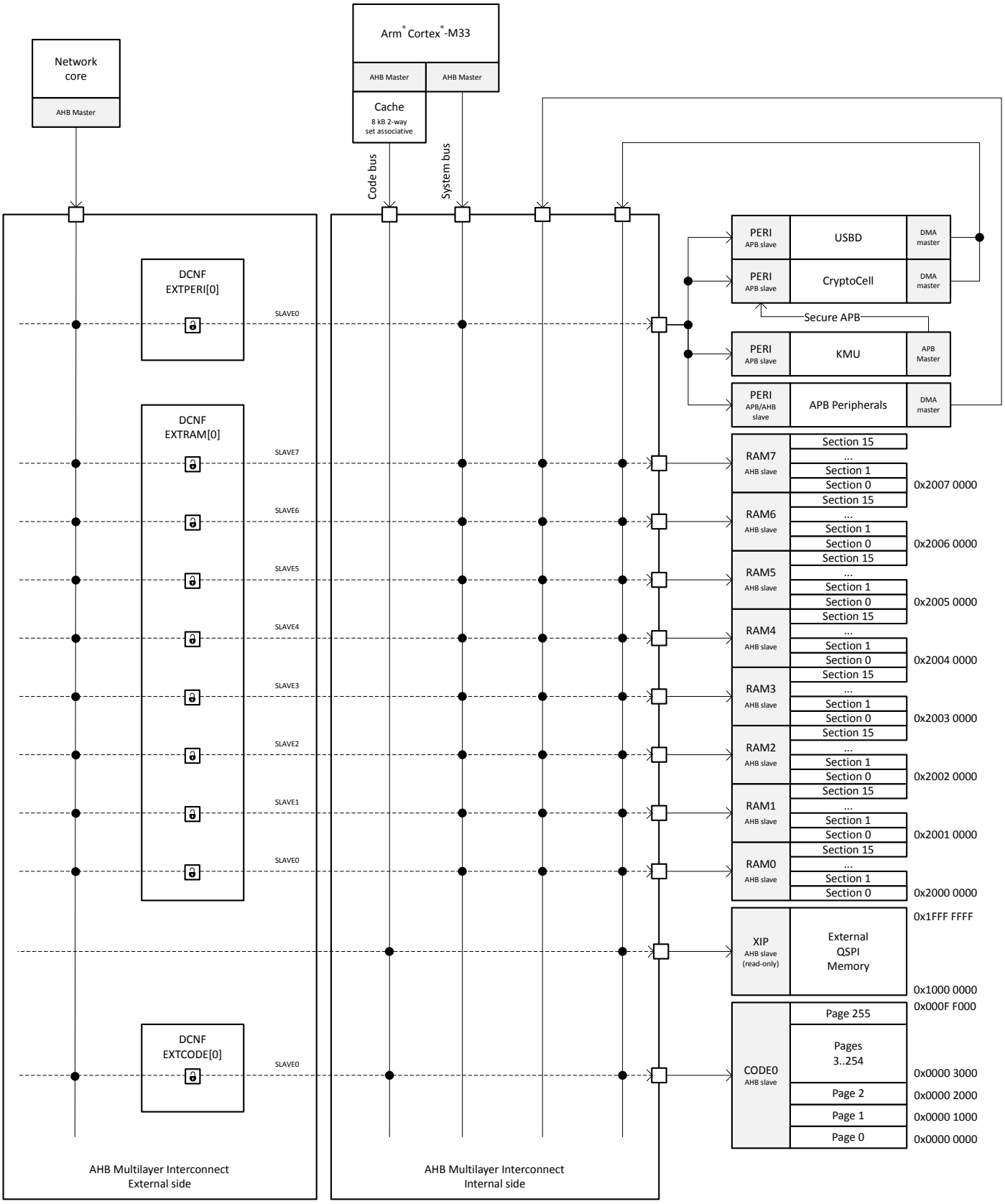

Figure 26: Memory layout

5.3.1 Peripheral instantiation

The following table describes the abbreviations used in the Instance, Secure mapping, and DMA security columns of the instantiation table.

Abbreviation	Description
NS	Non-secure - Peripheral is always accessible as a Non-Secure peripheral
S	Secure - Peripheral is always accessible as a Secure peripheral
US	User Selectable - A Secure or Non-secure attribute for the peripheral is defined in the SPU
SPLIT	Both Secure and Non-secure - The same resource is shared by both secure and non-secure code
NA	Not Applicable - Peripheral has no DMA capability
NSA	NoSeparateAttribute - Peripheral with DMA and DMA transfer has the same security attribute as assigned to the peripheral
SA	SeparateAttribute - Peripheral with DMA and DMA transfers can have a different security attribute than the one assigned to the peripheral

Table 29: Instantiation table abbreviations

The Secure mapping column in the following table defines configuration capabilities for the Arm TrustZone for Armv8-M secure attribute. The DMA security column describes the DMA capabilities of the peripheral.

5.3.1.1 Instantiation

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
0	0x50000000 0x40000000	DCNF	DCNF : S DCNF : NS	US	NA	Domain configuration
0	0x50000000 0x40000000	FPU	FPU : S FPU : NS	US	NA	Floating Point unit interrupt control
1	0x50001000	CACHE	CACHE	S	NA	Cache
3	0x50003000	SPU	SPU	S	NA	System protection unit
4	0x50004000 0x40004000	OSCILLATORS	OSCILLATORS : S OSCILLATORS : NS	US	NA	Oscillator configuration
4	0x50004000 0x40004000	REGULATORS	REGULATORS : S REGULATORS : NS	US	NA	Regulator configuration
5	0x50005000 0x40005000	CLOCK	CLOCK : S CLOCK : NS	US	NA	Clock control
5	0x50005000 0x40005000	POWER	POWER : S POWER : NS	US	NA	Power control
5	0x50005000 0x40005000	RESET	RESET : S RESET : NS	US	NA	Reset control and status
6	0x50006000 0x40006000	CTRLAPPERI	CTRLAP : S CTRLAP : NS	US	NSA	Control access port CPU side
8	0x50008000 0x40008000	SPIM	SPIM0 : S SPIM0 : NS	US	SA	SPI master 0
8	0x50008000 0x40008000	SPIS	SPIS0 : S SPIS0 : NS	US	SA	SPI slave 0
8	0x50008000 0x40008000	TWIM	TWIM0 : S TWIM0 : NS	US	SA	Two-wire interface master 0
8	0x50008000 0x40008000	TWIS	TWIS0 : S TWIS0 : NS	US	SA	Two-wire interface slave 0
8	0x50008000 0x40008000	UARTE	UARTE0 : S UARTE0 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 0

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
9	0x50009000 0x40009000	SPIM	SPIM1 : S SPIM1 : NS	US	SA	SPI master 1
9	0x50009000 0x40009000	SPIS	SPIS1 : S SPIS1 : NS	US	SA	SPI slave 1
9	0x50009000 0x40009000	TWIM	TWIM1 : S TWIM1 : NS	US	SA	Two-wire interface master 1
9	0x50009000 0x40009000	TWIS	TWIS1 : S TWIS1 : NS	US	SA	Two-wire interface slave 1
9	0x50009000 0x40009000	UARTE	UARTE1 : S UARTE1 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 1
10	0x5000A000 0x4000A000	SPIM	SPIM4 : S SPIM4 : NS	US	SA	SPI master 4 (high-speed)
11	0x5000B000 0x4000B000	SPIM	SPIM2 : S SPIM2 : NS	US	SA	SPI master 2
11	0x5000B000 0x4000B000	SPIS	SPIS2 : S SPIS2 : NS	US	SA	SPI slave 2
11	0x5000B000 0x4000B000	TWIM	TWIM2 : S TWIM2 : NS	US	SA	Two-wire interface master 2
11	0x5000B000 0x4000B000	TWIS	TWIS2 : S TWIS2 : NS	US	SA	Two-wire interface slave 2
11	0x5000B000 0x4000B000	UARTE	UARTE2 : S UARTE2 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 2
12	0x5000C000 0x4000C000	SPIM	SPIM3 : S SPIM3 : NS	US	SA	SPI master 3
12	0x5000C000 0x4000C000	SPIS	SPIS3 : S SPIS3 : NS	US	SA	SPI slave 3
12	0x5000C000 0x4000C000	TWIM	TWIM3 : S TWIM3 : NS	US	SA	Two-wire interface master 3
12	0x5000C000 0x4000C000	TWIS	TWIS3 : S TWIS3 : NS	US	SA	Two-wire interface slave 3
12	0x5000C000 0x4000C000	UARTE	UARTE3 : S UARTE3 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 3
13	0x5000D000	GPIOTE	GPIOTE0	S	NA	GPIO tasks and events
14	0x5000E000 0x4000E000	SAADC	SAADC : S SAADC : NS	US	SA	Successive approximation analog-to-digital converter
15	0x5000F000 0x4000F000	TIMER	TIMER0 : S TIMER0 : NS	US	NA	Timer 0
16	0x50010000 0x40010000	TIMER	TIMER1 : S TIMER1 : NS	US	NA	Timer 1
17	0x50011000 0x40011000	TIMER	TIMER2 : S TIMER2 : NS	US	NA	Timer 2
20	0x50014000 0x40014000	RTC	RTC0 : S RTC0 : NS	US	NA	Real time counter 0
21	0x50015000 0x40015000	RTC	RTC1 : S RTC1 : NS	US	NA	Real time counter 1
23	0x50017000 0x40017000	DPPIC	DPPIC : S DPPIC : NS	SPLIT	NA	DPPI controller
24	0x50018000 0x40018000	WDT	WDT0 : S WDT0 : NS	US	NA	Watchdog timer 0
25	0x50019000 0x40019000	WDT	WDT1 : S WDT1 : NS	US	NA	Watchdog timer 1
26	0x5001A000 0x4001A000	COMP	COMP : S COMP : NS	US	NA	Comparator

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
26	0x5001A000 0x4001A000	LPCOMP	LPCOMP : S LPCOMP : NS	US	NA	Low-power comparator
27	0x5001B000 0x4001B000	EGU	EGU0 : S EGU0 : NS	US	NA	Event generator unit 0
28	0x5001C000 0x4001C000	EGU	EGU1 : S EGU1 : NS	US	NA	Event generator unit 1
29	0x5001D000 0x4001D000	EGU	EGU2 : S EGU2 : NS	US	NA	Event generator unit 2
30	0x5001E000 0x4001E000	EGU	EGU3 : S EGU3 : NS	US	NA	Event generator unit 3
31	0x5001F000 0x4001F000	EGU	EGU4 : S EGU4 : NS	US	NA	Event generator unit 4
32	0x50020000 0x40020000	EGU	EGU5 : S EGU5 : NS	US	NA	Event generator unit 5
33	0x50021000 0x40021000	PWM	PWM0 : S PWM0 : NS	US	SA	Pulse width modulation unit 0
34	0x50022000 0x40022000	PWM	PWM1 : S PWM1 : NS	US	SA	Pulse width modulation unit 1
35	0x50023000 0x40023000	PWM	PWM2 : S PWM2 : NS	US	SA	Pulse width modulation unit 2
36	0x50024000 0x40024000	PWM	PWM3 : S PWM3 : NS	US	SA	Pulse width modulation unit 3
38	0x50026000 0x40026000	PDM	PDM0 : S PDM0 : NS	US	SA	Pulse density modulation (digital microphone) interface
40	0x50028000 0x40028000	I2S	I2S0 : S I2S0 : NS	US	SA	Inter-IC sound interface
42	0x5002A000 0x4002A000	IPC	IPC : S IPC : NS	US	NA	Interprocessor communication
43	0x5002B000 0x4002B000	QSPI	QSPI : S QSPI : NS	US	SA	External memory (quad serial peripheral) interface
45	0x5002D000 0x4002D000	NFCT	NFCT : S NFCT : NS	US	SA	Near field communication tag
47	0x4002F000	GPIOTE	GPIOTE1	NS	NA	GPIO tasks and events
48	0x50030000 0x40030000	MUTEX	MUTEX : S MUTEX : NS	US	NA	Mutual exclusive hardware support
51	0x50033000 0x40033000	QDEC	QDEC0 : S QDEC0 : NS	US	NA	Quadrature decoder 0
52	0x50034000 0x40034000	QDEC	QDEC1 : S QDEC1 : NS	US	NA	Quadrature decoder 1
54	0x50036000 0x40036000	USBD	USBD : S USBD : NS	US	SA	Universal serial bus device
55	0x50037000 0x40037000	USBREG	USBREGULATOR : S USBREGULATOR : NS	US	NA	USB regulator control
57	0x50039000 0x40039000	KMU	KMU : S KMU : NS	SPLIT	NA	Key management unit
57	0x50039000 0x40039000	NVMC	NVMC : S NVMC : NS	SPLIT	NA	Non-volatile memory controller
66	0x50842500 0x40842500	GPIO	P0 : S P0 : NS	SPLIT	NA	General purpose input and output, port 0
66	0x50842800 0x40842800	GPIO	P1 : S P1 : NS	SPLIT	NA	General purpose input and output, port 1

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
68	0x50844000	CRYPTOCELL	CRYPTOCELL	S	NSA	CryptoCell subsystem control interface
129	0x50081000 0x40081000	VMC	VMC : S VMC : NS	US	NA	Volatile memory controller
N/A	0x00F00000	CACHEDATA	CACHEDATA	S	NA	Cache data
N/A	0x00F08000	CACHEINFO	CACHEINFO	S	NA	Cache info
N/A	0x00FF0000	FICR	FICR	S	NA	Factory information configuration registers
N/A	0x00FF8000	UICR	UICR	S	NA	User information configuration registers
N/A	0xE0042000	CTI	CTI	S	NA	Cross-trigger interface
N/A	0xE0080000	TAD	TAD	S	NA	Trace and debug control

Table 30: Instantiation table

5.4 Core components

5.4.1 CACHE — Instruction/data cache

The cache is two-way set associative with a least recently used (LRU) replacement policy. Both instruction and data accesses towards flash memory or XIP code regions are cached.

The cache has the following features:

- 128-bit cache line
- Configurable as a cache or general purpose RAM
- Hit/miss counters per NVM region and access type (instruction or data)
- Readable cache content (for profiling)
  - Data, tag, valid, and most recently used (MRU) bits
  - Can be disabled when not in use
- Manual invalidation and erase support
- Locking cache updates on cache misses

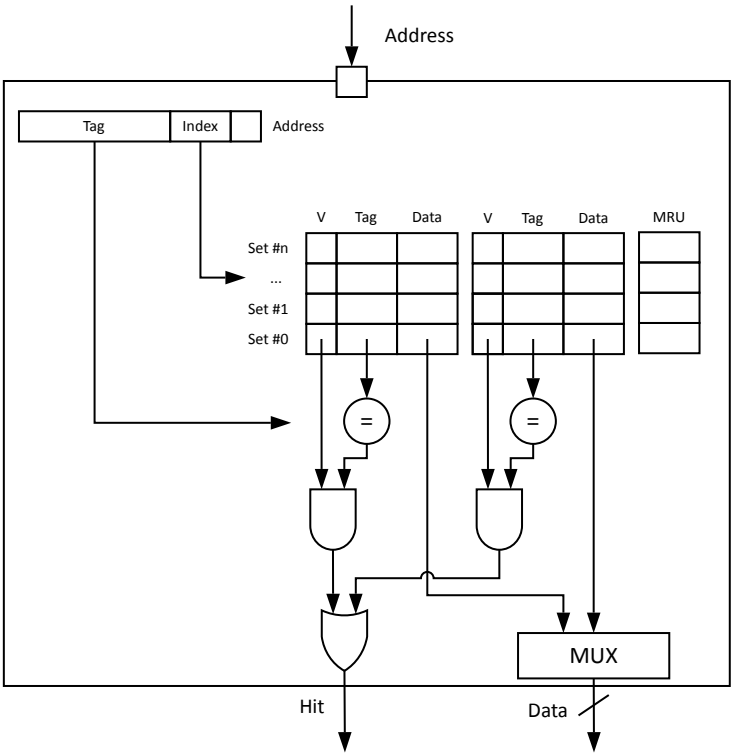

Figure 27: Cache overview

In Cache mode (MODE=Cache), instruction and data accesses from the CPU over the code bus towards internal or external flash, are cached. The contents of the cache, i.e. data, tag, valid, and MRU bits, are memory mapped, see Cache content on page 109. This can be used for performance profiling of code running in the system. Access to the cache content region is read-only by default, but can be blocked by enabling a lock bit in DEBUGLOCK on page 114. Preventing cache content updates on cache misses can be enabled through register DEBUGLOCK on page 114. When enabled, cache content is not replaced, but kept intact. The cache is still enabled and provides fast instruction and data fetches for cached content.

Access to internal or external flash memory will not be cached when in Ram mode (MODE=Ram). Instead, the cache data content, as described in Cache content on page 109, can be used as read/write RAM.

5.4.1.1 Cache content

Cache information is divided into cache info content and cache data content.

Cache info content is organized in memory as shown in the following figure.

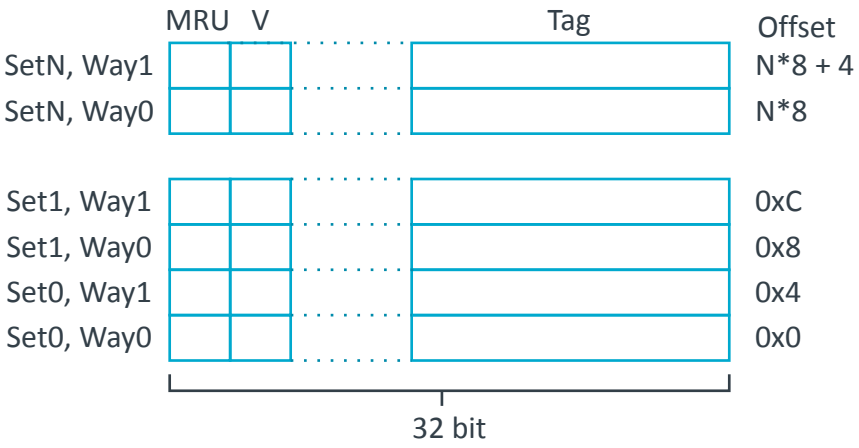

Figure 28: Cache info content

The V field contains the bit that indicates if a cache entry is valid or not. All V fields are cleared when invalidating the cache using register INVALIDATE on page 113, when disabling the cache using register ENABLE on page 112, or when changing MODE from `Ram` to `Cache`. The MRU field indicates which way was used most recently in the set. The MRU bit is updated on each fetch from the cache and is used for the cache replacement policy. The Tag field is used to check if an entry in the cache matches the address being fetched.

The following figure shows how the cache data content is organized in memory.

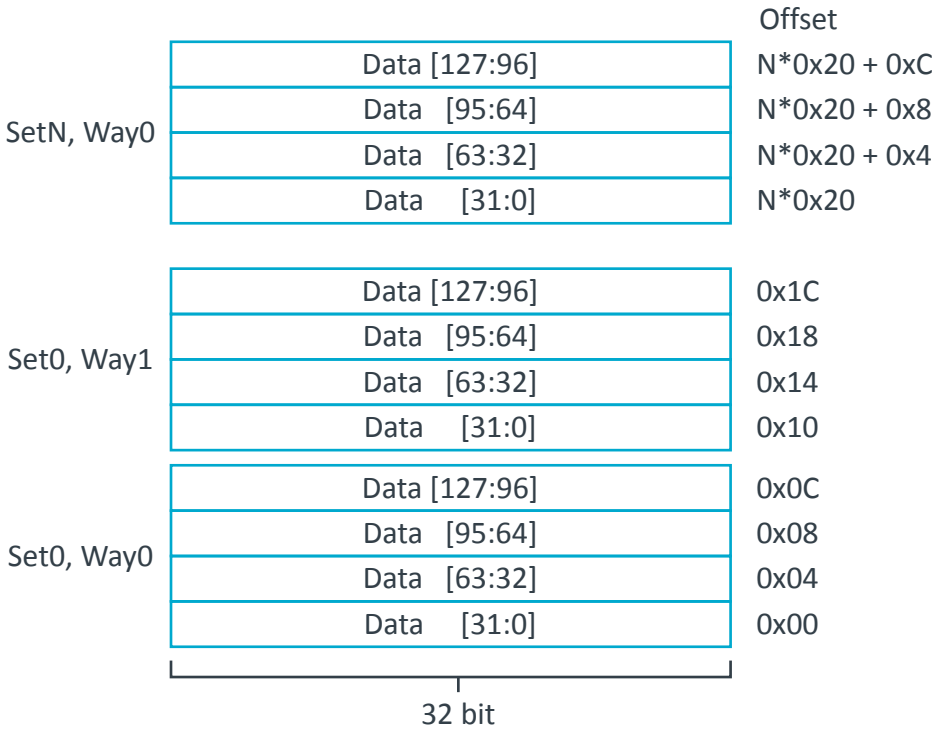

Figure 29: Cache data content

Each set consists of two ways, each containing 128 bits of data. The 128-bit data is available as 4x32-bit words in sequential order. When operating in Ram mode (MODE = `Ram`), the data is accessible as general purpose RAM.

The cache info and cache data content are memory mapped in the CACHEINFO and CACHEDATA regions. These can be accessed in the CACHEINFO registers and CACHEDATA registers respectively.

5.4.1.2 Profiling

The cache includes profiling counters IHIT, IMISS, DHIT, and DMISS for both flash and execute-in-place (XIP).

Cache performance on executed code is indicated by these counters when enabled through PROFILINGENABLE on page 113. The counters can be cleared at any time using PROFILINGCLEAR on page 113. Writing to this register will clear all profiling counters. After being cleared, the counters will increment at the next instruction, or data fetch, according to the rules in the following table.

Profiling counter	Description
IHIT	Increased on a cache hit for instruction fetch
IMISS	Increased on a cache miss for instruction fetch
DHIT	Increased on a cache hit for data fetch (i.e. LOAD type instruction targeting the cache region)
DMISS	Increased on a cache miss for data fetch (i.e. LOAD type instruction targeting the cache region)

Table 31: Profiling counters

5.4.1.3 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50001000	APPLICATION	CACHE	CACHE	S	NA	Cache	

Table 32: Instances

Register	Offset	Security	Description
PROFILING[n].IHIT	0x400		Instruction fetch cache hit counter for cache region n, where n=0 means Flash and n=1 means XIP.
PROFILING[n].IMISS	0x404		Instruction fetch cache miss counter for cache region n, where n=0 means Flash and n=1 means XIP.
PROFILING[n].DHIT	0x408		Data fetch cache hit counter for cache region n, where n=0 means Flash and n=1 means XIP.
PROFILING[n].DMISS	0x40C		Data fetch cache miss counter for cache region n, where n=0 means Flash and n=1 means XIP.
ENABLE	0x500		Enable cache.
INVALIDATE	0x504		Invalidate the cache.
ERASE	0x508		Erase the cache.
PROFILINGENABLE	0x50C		Enable the profiling counters.
PROFILINGCLEAR	0x510		Clear the profiling counters.
MODE	0x514		Cache mode. Switching from Cache to Ram mode causes the RAM to be cleared. Switching from RAM to Cache mode causes the cache to be invalidated.
DEBUGLOCK	0x518		Lock debug mode.
ERASESTATUS	0x51C		Cache erase status.
WRITELOCK	0x520		Lock cache updates. Prevents updating of cache content on cache misses, but will continue to lookup instruction/data fetches in content already present in the cache. Ignored in RAM mode.

Table 33: Register overview

5.4.1.3.1 PROFILING[n].IHIT (n=0..1)

Address offset: 0x400 + (n × 0x20)

Instruction fetch cache hit counter for cache region n, where n=0 means Flash and n=1 means XIP.

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

5.4.1.3.2 PROFILING[n].IMISS (n=0..1)

Address offset: 0x404 + (n × 0x20)

Instruction fetch cache miss counter for cache region n, where n=0 means Flash and n=1 means XIP.

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

5.4.1.3.3 PROFILING[n].DHIT (n=0..1)

Address offset: 0x408 + (n × 0x20)

Data fetch cache hit counter for cache region n, where n=0 means Flash and n=1 means XIP.

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

5.4.1.3.4 PROFILING[n].DMISS (n=0..1)

Address offset: 0x40C + (n × 0x20)

Data fetch cache miss counter for cache region n, where n=0 means Flash and n=1 means XIP.

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

5.4.1.3.5 ENABLE

Address offset: 0x500

Enable cache.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable cache
			Disabled	0	Disable cache
			Enabled	1	Enable cache

5.4.1.3.6 INVALIDATE

Address offset: 0x504

Invalidate the cache.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	INVALIDATE			Invalidate the cache
			Invalidate	1	Invalidate the cache

5.4.1.3.7 ERASE

Address offset: 0x508

Erase the cache.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	ERASE			Erase the cache
			Erase	1	Erase cache

5.4.1.3.8 PROFILINGENABLE

Address offset: 0x50C

Enable the profiling counters.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable the profiling counters
			Disable	0	Disable profiling
			Enable	1	Enable profiling

5.4.1.3.9 PROFILINGCLEAR

Address offset: 0x510

Clear the profiling counters.

The profiling counters can be cleared at any time. When cleared, all profiling counters will be set to zero, and will increment at the next instruction or data fetch.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	CLEAR			Clearing the profiling counters
			Clear	1	Clear the profiling counters

5.4.1.3.10 MODE

Address offset: 0x514

Cache mode.

Switching from Cache to Ram mode causes the RAM to be cleared.

Switching from RAM to Cache mode causes the cache to be invalidated.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	MODE			Cache mode
			Cache	0	Cache mode
			Ram	1	RAM mode

5.4.1.3.11 DEBUGLOCK

Address offset: 0x518

Lock debug mode.

This register is ignored when CACHE is used in RAM mode. Once this register has been set to Locked, the debug mode can only be unlocked by resetting the device.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW1	DEBUGLOCK			Lock debug mode
			Unlocked	0	Debug mode unlocked
			Locked	1	Debug mode locked

5.4.1.3.12 ERASESTATUS

Address offset: 0x51C

Cache erase status.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ERASESTATUS			Cache erase status
			Idle	0	Erase is not complete or hasn't started
			Finished	1	Cache erase is finished Write 0 to clear.

5.4.1.3.13 WRITELOCK

Address offset: 0x520

Lock cache updates. Prevents updating of cache content on cache misses, but will continue to lookup instruction/data fetches in content already present in the cache.

Ignored in RAM mode.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	WRITELOCK			Lock cache updates
			Unlocked	0	Cache updates unlocked
			Locked	1	Cache updates locked

5.4.1.4 Electrical specification

5.4.1.4.1 Cache size

Symbol	Description	Min.	Typ.	Max.	Units
$Size_{CACHEDATA}$	CACHEDATA size		8192		Bytes

5.4.1.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x00F08000	APPLICATION	CACHEINFO	CACHEINFO	S	NA	Cache info	

Table 34: Instances

Register	Offset	Security	Description
SET[n].WAY[o]	0x0		Cache information for SET[n], WAY[o].

Table 35: Register overview

5.4.1.5.1 SET[n].WAY[o] (n=0..255) (o=0..1)

Address offset: 0x0 + (n × 0x8) + (o × 0x4)

Cache information for SET[n], WAY[o].

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	TAG			Cache tag.
B	R	V			Valid bit
			Invalid	0	Invalid cache line
			Valid	1	Valid cache line
C	R	MRU			Most recently used way.
			Way0	0	Way0 was most recently used
			Way1	1	Way1 was most recently used

5.4.1.6 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x00F00000	APPLICATION	CACHEDATA	CACHEDATA	S	NA	Cache data	

Table 36: Instances

Register	Offset	Security	Description
SET[n].WAY[o].DATA0	0x0		Cache data bits [31:0] of SET[n], WAY[o].
SET[n].WAY[o].DATA1	0x4		Cache data bits [63:32] of SET[n], WAY[o].
SET[n].WAY[o].DATA2	0x8		Cache data bits [95:64] of SET[n], WAY[o].
SET[n].WAY[o].DATA3	0xC		Cache data bits [127:96] of SET[n], WAY[o].

Table 37: Register overview

5.4.1.6.1 SET[n].WAY[o].DATA0 (n=0..255) (o=0..1)

Address offset: 0x0 + (n × 0x20) + (o × 0x10)

Cache data bits [31:0] of SET[n], WAY[o].

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

5.4.1.6.2 SET[n].WAY[o].DATA1 (n=0..255) (o=0..1)

Address offset: 0x4 + (n × 0x20) + (o × 0x10)

Cache data bits [63:32] of SET[n], WAY[o].

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

5.4.1.6.3 SET[n].WAY[o].DATA2 (n=0..255) (o=0..1)

Address offset: 0x8 + (n × 0x20) + (o × 0x10)

Cache data bits [95:64] of SET[n], WAY[o].

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

5.4.1.6.4 SET[n].WAY[o].DATA3 (n=0..255) (o=0..1)

Address offset: 0xC + (n × 0x20) + (o × 0x10)

Cache data bits [127:96] of SET[n], WAY[o].

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

5.4.2 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.

Note: FICR is not accessible from the network core.

5.4.2.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x00FF0000	APPLICATION	FICR	FICR	S	NA	Factory information configuration registers	

Table 38: Instances

Register	Offset	Security	Description
INFO.CONFIGID	0x200		Configuration identifier
INFO.DEVICEID[n]	0x204		Device identifier
INFO.PART	0x20C		Part code
INFO.VARIANT	0x210		Part Variant, Hardware version and Production configuration
INFO.PACKAGE	0x214		Package option
INFO.RAM	0x218		RAM variant
INFO.FLASH	0x21C		Flash variant
INFO.CODEPAGESIZE	0x220		Code memory page size in bytes
INFO.CODESIZE	0x224		Code memory size
INFO.DEVICETYPE	0x228		Device type
TRIMCNF[n].ADDR	0x300		Address of the PAR register which will be written
TRIMCNF[n].DATA	0x304		Data
NFC.TAGHEADER0	0x450		Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER1	0x454		Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER2	0x458		Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
NFC.TAGHEADER3	0x45C		Default header for NFC Tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST.
TRNG90B.BYTES	0xC00		Amount of bytes for the required entropy bits
TRNG90B.RCCUTOFF	0xC04		Repetition counter cutoff
TRNG90B.APCUTOFF	0xC08		Adaptive proportion cutoff
TRNG90B.STARTUP	0xC0C		Amount of bytes for the startup tests
TRNG90B.ROSC1	0xC10		Sample count for ring oscillator 1
TRNG90B.ROSC2	0xC14		Sample count for ring oscillator 2
TRNG90B.ROSC3	0xC18		Sample count for ring oscillator 3
TRNG90B.ROSC4	0xC1C		Sample count for ring oscillator 4

Register	Offset	Security	Description
XOSC32MTRIM	0xC20		XOSC32M capacitor selection trim values

Table 39: Register overview

5.4.2.1.1 INFO.CONFIGID

Address offset: 0x200

Configuration 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
Reset 0xFFFFFFFF	**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	R/W	Field	Value ID	Value	Description
A	R	HWID			Identification number for the HW

5.4.2.1.2 INFO.DEVICEID[n] (n=0..1)

Address offset: 0x204 + (n × 0x4)

Device identifier

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

ID	R/W	Field	Value ID	Value	Description
A	R	DEVICEID			64 bit unique device identifier DEVICEID[0] contains the least significant bits of the device identifier. DEVICEID[1] contains the most significant bits of the device identifier.

5.4.2.1.3 INFO.PART

Address offset: 0x20C

Part code

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00005340	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	R	PART			Part code
			N5340	0x5340	nRF5340
			Unspecified	0xFFFFFFFF	Unspecified

5.4.2.1.4 INFO.VARIANT

Address offset: 0x210

Part Variant, Hardware version and Production configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	VARIANT			Part Variant, Hardware version and Production configuration, encoded as ASCII
			QKAA	0x514B4141	QKAA
			Unspecified	0xFFFFFFFF	Unspecified

5.4.2.1.5 INFO.PACKAGE

Address offset: 0x214

Package option

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	PACKAGE			Package option
			QK	0x2000	QKxx - 94-pin aQFN
			Unspecified	0xFFFFFFFF	Unspecified

5.4.2.1.6 INFO.RAM

Address offset: 0x218

RAM variant

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	RAM			RAM variant
			K16	0x10	16 kByte RAM
			K32	0x20	32 kByte RAM
			K64	0x40	64 kByte RAM
			K128	0x80	128 kByte RAM
			K256	0x100	256 kByte RAM
			K512	0x200	512 kByte RAM
			Unspecified	0xFFFFFFFF	Unspecified

5.4.2.1.7 INFO.FLASH

Address offset: 0x21C

Flash variant

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

5.4.2.1.8 INFO.CODEPAGESIZE

Address offset: 0x220

Code memory page size in bytes

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00001000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	CODEPAGESIZE			Code memory page size in bytes
			K4096	0x1000	4 kByte

5.4.2.1.9 INFO.CODESIZE

Address offset: 0x224

Code memory size

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000100				**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	R/W	Field	Value ID	Value	Description
A	R	CODESIZE			Code memory size in number of pages Total code space is: CODEPAGESIZE * CODESIZE bytes
			P256	256	256 pages

5.4.2.1.10 INFO.DEVICETYPE

Address offset: 0x228

Device 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	DEVICETYPE			Device type
			Die	0x0000000	Device is an physical DIE
			FPGA	0xFFFFFFFF	Device is an FPGA

5.4.2.1.11 TRIMCNF[n].ADDR (n=0..31)

Address offset: 0x300 + (n × 0x8)

Address of the PAR register which will be written

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	R/W	Field																														
A	R	Address																														

5.4.2.1.12 TRIMCNF[n].DATA (n=0..31)

Address offset: 0x304 + (n × 0x8)

Data

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	R/W	Field																														
A	R	Data																														

5.4.2.1.13 NFC.TAGHEADER0

Address offset: 0x450

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

Bit number																																
ID																																
Reset 0xFFFFFF5F																																
ID	R/W	Field																														
A	R	MFGID																														
B	R	UD1																														
C	R	UD2																														
D	R	UD3																														

5.4.2.1.14 NFC.TAGHEADER1

Address offset: 0x454

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

Bit number																																
ID																																
Reset 0xFFFFFFFF																																
ID	R/W	Field																														
A-D	R	UD[i] (i=4..7)																														

5.4.2.1.15 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	D	C	C	C	C	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	R/W	Field	Value ID	Value								Description																							
A-D	R	UD[i] (i=8..11)										Unique identifier byte i																							

5.4.2.1.16 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	D	C	C	C	C	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	R/W	Field	Value ID	Value								Description																							
A-D	R	UD[i] (i=12..15)										Unique identifier byte i																							

5.4.2.1.17 TRNG90B.BYTES

Address offset: 0xC00

Amount of bytes for the required entropy bits

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000210				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
A	R	BYTES										Amount of bytes for the required entropy bits																							

5.4.2.1.18 TRNG90B.RCCUTOFF

Address offset: 0xC04

Repetition counter cutoff

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	R/W	Field	Value ID	Value								Description																							
A	R	RCCUTOFF										Repetition counter cutoff																							

5.4.2.1.19 TRNG90B.APCUTOFF

Address offset: 0xC08

Adaptive proportion cutoff

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	APCUTOFF			Adaptive proportion cutoff

5.4.2.1.20 TRNG90B.STARTUP

Address offset: 0xC0C

Amount of bytes for the startup tests

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	STARTUP			Amount of bytes for the startup tests

5.4.2.1.21 TRNG90B.ROSC1

Address offset: 0xC10

Sample count for ring oscillator 1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	ROSC1			Sample count for ring oscillator 1

5.4.2.1.22 TRNG90B.ROSC2

Address offset: 0xC14

Sample count for ring oscillator 2

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	ROSC2			Sample count for ring oscillator 2

5.4.2.1.23 TRNG90B.ROSC3

Address offset: 0xC18

Sample count for ring oscillator 3

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	ROSC3			Sample count for ring oscillator 3

5.4.2.1.24 TRNG90B.ROSC4

Address offset: 0xC1C

Sample count for ring oscillator 4

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	ROSC4			Sample count for ring oscillator 4

5.4.2.1.25 XOSC32MTRIM

Address offset: 0xC20

XOSC32M capacitor selection trim values

Note: To enable the optional internal capacitors on XC1 and XC2 pins, see to the "Using internal capacitors" section of the OSCILLATORS chapter.

Bit number				31 30 29 28 27 26 25 24 23 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 A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	SLOPE		[-16..15]	Slope trim factor on twos complement form -16: Minimum slope 15: Maximum slope
B	R	OFFSET		[31..0]	Offset trim factor on integer form 0: Minimum offset 31: Maximum offset

5.4.3 UICR — User information configuration registers

The user information configuration registers (UICRs) are non-volatile memory (NVM) registers for configuring user specific settings and storing secure cryptographic keys or OTP values.

The cryptographic key part of the UICR (addresses starting at 0x100 and higher) is handled by the Key Management Unit (KMU), see KMU — Key management unit on page 270 for more information.

For information on writing registers, see NVMC — Non-volatile memory controller on page 329 and Memory on page 18.

Note: UICR is not accessible from the network core.

5.4.3.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x00FF8000	APPLICATION	UICR	UICR	S	NA	User information configuration registers	

Table 40: Instances

Register	Offset	Security	Description
APPROTECT	0x000		Access port protection
VREGHVOUT	0x010		Output voltage from the high voltage (VREGH) regulator stage. The maximum output voltage from this stage is given as VDDH - VREGHDROP.
HFXOCNT	0x014		HFXO startup counter
SECUREAPPROTECT	0x01C		Secure access port protection
ERASEPROTECT	0x020		Erase protection
TINSTANCE	0x024		SW-DP Target instance
NFCPINS	0x028		Setting of pins dedicated to NFC functionality: NFC antenna or GPIO
OTP[n]	0x100		One time programmable memory
KEYSLOT.CONFIG[n].DEST	0x400		Destination address where content of the key value registers (KEYSLOT.KEYn.VALUE[0-3]) will be pushed by KMU. Note that this address must match that of a peripherals APB mapped write-only key registers, else the KMU can push this key value into an address range which the CPU can potentially read.
KEYSLOT.CONFIG[n].PERM	0x404		Define permissions for the key slot. Bits 0-15 and 16-31 can only be written when equal to 0xFFFF.
KEYSLOT.KEY[n].VALUE[o]	0x800		Define bits [31+o*32:0+o*32] of value assigned to KMU key slot.

Table 41: Register overview

5.4.3.1.1 APPROTECT

Address offset: 0x000

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	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	R/W	Field	Value ID	Value	Description																														
A	RW	PALL			Blocks debugger read/write access to all CPU registers and memory mapped addresses. Using any value except Unprotected will lead to the protection being enabled. Bits with value '1' can be written to '0'. Bits with value '0' cannot be written to '1'.																														
			Unprotected	0x50FA50FA	Unprotected																														
			Protected	0x00000000	Protected																														

5.4.3.1.2 VREGHVOUT

Address offset: 0x010

Output voltage from the high voltage (VREGH) regulator stage. The maximum output voltage from this stage is given as VDDH - VREGHDROP.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	VREGHVOUT			VREGH regulator output voltage.
			1V8	0	1.8 V
			2V1	1	2.1 V
			2V4	2	2.4 V
			2V7	3	2.7 V
			3V0	4	3.0 V
			3V3	5	3.3 V
			DEFAULT	7	Default voltage: 1.8 V

5.4.3.1.3 HFXOCNT

Address offset: 0x014

HFXO startup counter

The CLKSTARTED events are generated after the HFXO power up time + the HFXOCNT-defined debounce time + PLL lock time has elapsed. If the HFXO has already been requested by another clock source and is already running, the CLKSTARTED event is generated as soon as the PLL has locked.

Note: When HFXOCNT field of this register is 0xFF, e.g. after UICR being erased, a debounce time of (4*64 us + 0.5 us) is used

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

ID	R/W	Field	Value ID	Value	Description
A	RW	HFXOCNT			HFXO startup counter. Total debounce time = HFXOCNT*64 us + 0.5 us
			MinDebounceTime	0	Min debounce time = (0*64 us + 0.5 us)
			MaxDebounceTime	254	Max debounce time = (254*64 us + 0.5 us)
			DefaultDebounceTime	255	Default debounce time for erased UICR = 4*64 us + 0.5 us

5.4.3.1.4 SECUREAPPROTECT

Address offset: 0x01C

Secure 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 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	R/W	Field	Value ID	Value	Description
A	RW	PALL			Blocks debugger read/write access to all secure CPU registers and secure memory mapped addresses. Using any value except Unprotected will lead to the protection being enabled. Bits with value '1' can be written to '0'. Bits with value '0' cannot be written to '1'.
			Unprotected	0x50FA50FA	Unprotected
			Protected	0x00000000	Protected

5.4.3.1.5 ERASEPROTECT

Address offset: 0x020

Erase 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 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	R/W	Field	Value ID	Value	Description
A	RW	PALL			Blocks NVMC ERASEALL and CTRLAP ERASEALL functionality. Using any value except Unprotected will lead to the protection being enabled.
			Unprotected	0xFFFFFFFF	Unprotected
			Protected	0x00000000	Protected

5.4.3.1.6 TINSTANCE

Address offset: 0x024

SW-DP Target instance

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 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 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	TINSTANCE			TINSTANCE bits are negated and used in the SW-DP DLPIDR.TINSTANCE field. E.g. 0xF in this field is translated to 0x0 in DLPIDR.TINSTANCE field.

5.4.3.1.7 NFCPINS

Address offset: 0x028

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

Note: When used as NFC antenna pin, the corresponding pin must be controlled by the application core, and the GPIO PIN_CNF register initialized to its reset 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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	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

5.4.3.1.8 OTP[n] (n=0..191)

Address offset: 0x100 + (n × 0x4)

One time programmable 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				B B B B B B B B B B B B B B B B A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW1	LOWER			Lower half word **Note:** Can only be written to a non 0xFFFF value once.
B	RW1	UPPER			Upper half word **Note:** Can only be written to a non 0xFFFF value once.

5.4.3.1.9 KEYSLOT.CONFIG[n].DEST (n=0..127)

Address offset: 0x400 + (n × 0x8)

Destination address where content of the key value registers (KEYSLOT.KEYn.VALUE[0-3]) will be pushed by KMU. Note that this address must match that of a peripherals APB mapped write-only key registers, else the KMU can push this key value into an address range which the CPU can potentially read.

Note: Writing/reading this register requires the KMU SELECTKEYSLOT register to be set to n+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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	DEST			Secure APB destination address

5.4.3.1.10 KEYSLOT.CONFIG[n].PERM (n=0..127)

Address offset: 0x404 + (n × 0x8)

Define permissions for the key slot. Bits 0-15 and 16-31 can only be written when equal to 0xFFFF.

Note: Writing/reading this register requires the KMU SELECTKEYSLOT register to be set to n+1.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	WRITE			Write permission for key slot
			Disabled	0	Disable write to the key value registers
			Enabled	1	Enable write to the key value registers
B	RW	READ			Read permission for key slot
			Disabled	0	Disable read from key value registers
			Enabled	1	Enable read from key value registers
C	RW	PUSH			Push permission for key slot
			Disabled	0	Disable pushing of key value registers over secure APB, but can be read if field READ is Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Enable pushing of key value registers over secure APB. Register KEYSLOT.CONFIGn.DEST must contain a valid destination address!
D	RW	STATE			Revocation state for the key slot Note that it is not possible to undo a key revocation by writing the value '1' to this field
			Revoked	0	Key value registers can no longer be read or pushed
			Active	1	Key value registers are readable (if enabled) and can be pushed (if enabled)

5.4.3.1.11 KEYSLOT.KEY[n].VALUE[o] (n=0..127) (o=0..3)

Address offset: 0x800 + (n × 0x10) + (o × 0x4)

Define bits [31+o*32:0+o*32] of value assigned to KMU key slot.

Note: Writing/reading this register requires the KMU SELECTKEYSLOT register to be set to n+1.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Define bits [31+o*32:0+o*32] of value assigned to KMU key slot

5.4.4 AHB multilayer

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

Each bus master is connected to a slave device through one or more interconnection matrixes. The bus masters are assigned priorities that are used to resolve access when two (or more) bus masters request access to the same slave device. The following applies when assigning priorities:

- If two (or more) bus masters request access to the same slave device, the master with the highest priority is granted access first.
- If a higher priority bus master transaction is on-going, bus masters with lower priority are stalled. Either until the higher priority master has completed its transaction, or the higher priority master at any point pauses during its transaction. During this pause, the lower priority master in queue is temporarily granted access to the slave device until the higher priority master resumes its activity.
- If a lower priority bus master transaction is on-going, and a master with higher priority requests access to the same slave device, the lower priority bus master is stalled after the current word access has been completed. The slave transaction will continue following the rules above.
- Bus masters that have the same priority are mutually exclusive, and cannot be used concurrently.

Some peripherals, like I2S, do not have a safe stalling mechanism (not able to pause incoming data and no internal data buffering). Being a low priority bus master might cause loss of data for such peripherals upon bus contention. To avoid AHB bus contention when using multiple bus masters, apply one of the following guidelines:

- Avoid situations where more than one bus master is accessing the same slave.

- If more than one bus master is accessing the same slave, make sure that the bus bandwidth is not exhausted.

5.4.4.1 AHB multilayer priorities

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

Bus master name	Natural relative priority	In/Out
CPU	Highest priority	I/O
Network core		I/O
I2S		I/O
PDM		I
UARTE0, SPIM0, SPIS0, TWIM0, TWIS0		I/O
UARTE1, SPIM1, SPIS1, TWIM1, TWIS1		I/O
UARTE2, SPIM2, SPIS2, TWIM2, TWIS2		I/O
UARTE3, SPIM3, SPIS3, TWIM3, TWIS3		I/O
SAADC		I
PWM0		O
PWM1		O
PWM2		O
PWM3		O
SPIM4		I/O
NFCT		I/O
CC312		I/O
USBD		I/O
QSPI	Lowest priority	I/O

Table 42: AHB bus masters

6 Network core

The network core contains a low-power microcontroller with embedded flash memory and an Arm Cortex-M33 processor.

The network core includes peripherals for efficient implementation of radio protocols such as Bluetooth Low Energy, 802.15.4, and proprietary 2.4 GHz. The following figure provides more information. Arrows with white heads indicate signals that share physical pins with other signals.

6.1 Block diagram

The following image shows the network core block diagram.

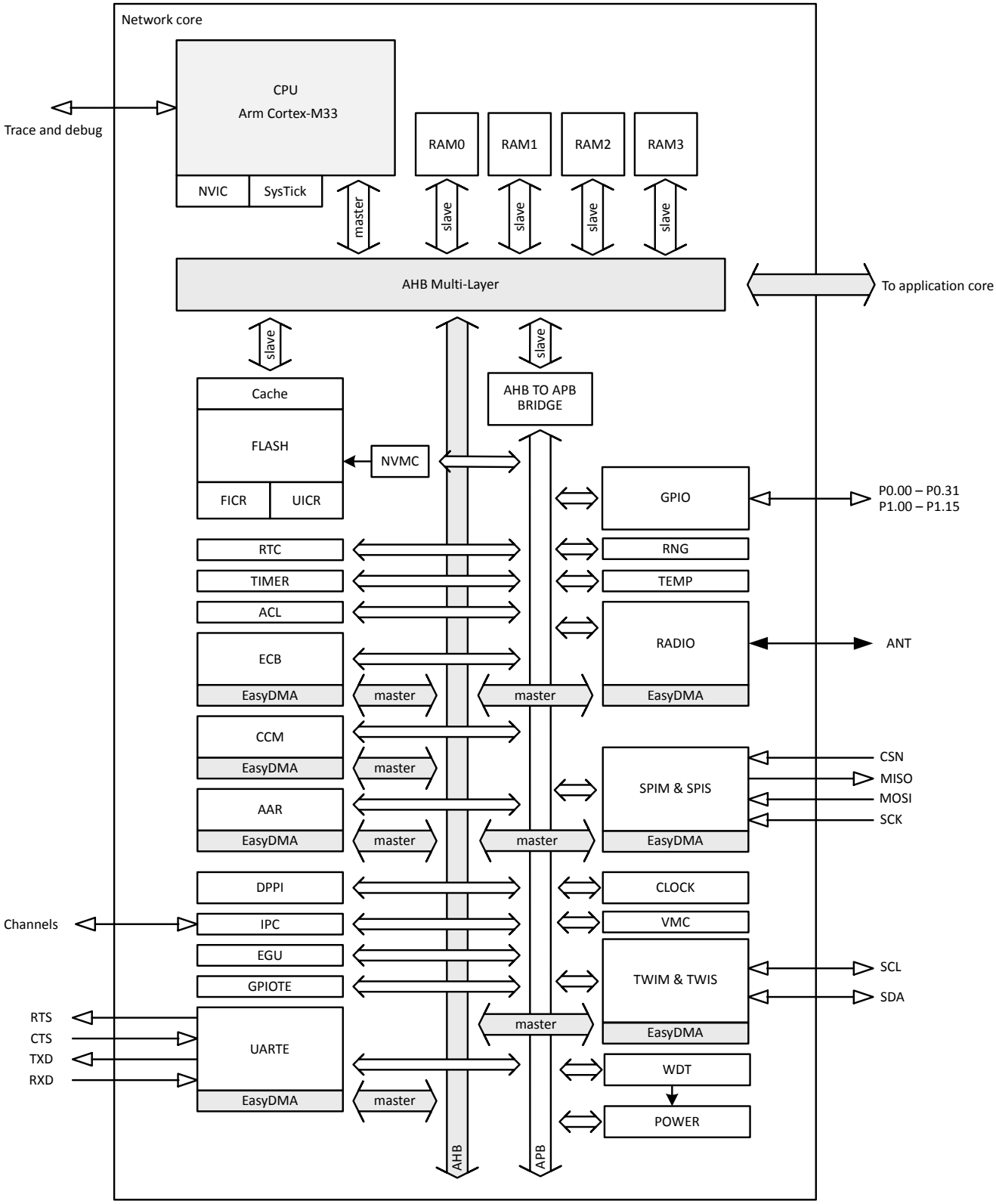

Figure 30: Network core block diagram

6.2 CPU

The Arm Cortex-M33 processor has a 32-bit instruction set (Thumb-2 technology) that implements a superset of 16- and 32-bit instructions to maximize code density and performance.

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

- Hardware divide
- 8- and 16-bit single instruction multiple data (SIMD) instructions
- Memory Protection Unit (MPU)

The Arm Cortex Microcontroller Software Interface Standard (CMSIS) is implemented and available for the application processor.

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 internal or external flash will have a wait state penalty. The instruction cache can be enabled to minimize flash wait states when fetching instructions. For more information on cache, see NVMC — Non-volatile memory controller on page 329. CPU performance parameters including mode wait states, CPU current and efficiency, and processing power and efficiency based on the CoreMark benchmark can be found in Electrical specification on page 133.

6.2.1 Electrical specification

6.2.1.1 CPU performance

Symbol	Description	Min.	Typ.	Max.	Units
W_{FLASH}	CPU wait states, running from flash, cache disabled	0		4	
$W_{FLASHCACHE}$	CPU wait states, running from flash, cache enabled	0		5	
W_{RAM}	CPU wait states, running from RAM		0		
$CM_{FLASHCACHE}$	CoreMark, running from flash, cache enabled		244		CoreMark
$CM_{FLASH/MHz}$	CoreMark per MHz, running from flash, cache enabled		3.8		CoreMark/MHz
$CM_{FLASH/mA}$	CoreMark per mA, running from flash, cache enabled		101		CoreMark/mA

6.2.2 CPU and support module configuration

The Arm Cortex-M33 processor has a number of CPU options and support modules implemented on the device.

Option/Module	Description	Implemented
Core options		
PRIORITIES	Priority bits	3
WIC	Wakeup Interrupt Controller	NO
Endianness	Memory system endianness	Little endian
DWT	Data Watchpoint and Trace	YES
Modules		
MPU	Number of non-secure MPU regions	8
	Number of secure MPU regions	0 (No Armv8-M Security Extensions)
SAU	Number of SAU regions	0 (No Arm TrustZone for Armv8-M Security Extensions)
FPU	Floating-point unit	NO
DSP	Digital Signal Processing Extension	NO
Armv8-M TrustZone	Arm TrustZone for Armv8-M Security Extensions	NO
CPIF	Coprocessor interface	NO
ETM	Embedded Trace Macrocell	NO
ITM	Instrumentation Trace Macrocell	NO
MTB	Micro Trace Buffer	NO
CTI	Cross Trigger Interface	YES
BPU	Breakpoint Unit	YES
HTM	AHB Trace Macrocell	NO

6.3 Memory

The network core contains flash memory and RAM that can be used for code and data storage.

The following figure shows how the CPU and peripherals with EasyDMA can access RAM via the AHB multilayer interconnect.

The network core can access application core resources (flash, RAM, and peripherals) when granted permission through the application's DCNF and SPU settings. A small portion of the application core RAM is dedicated to the exchange of messages between the application and network cores.

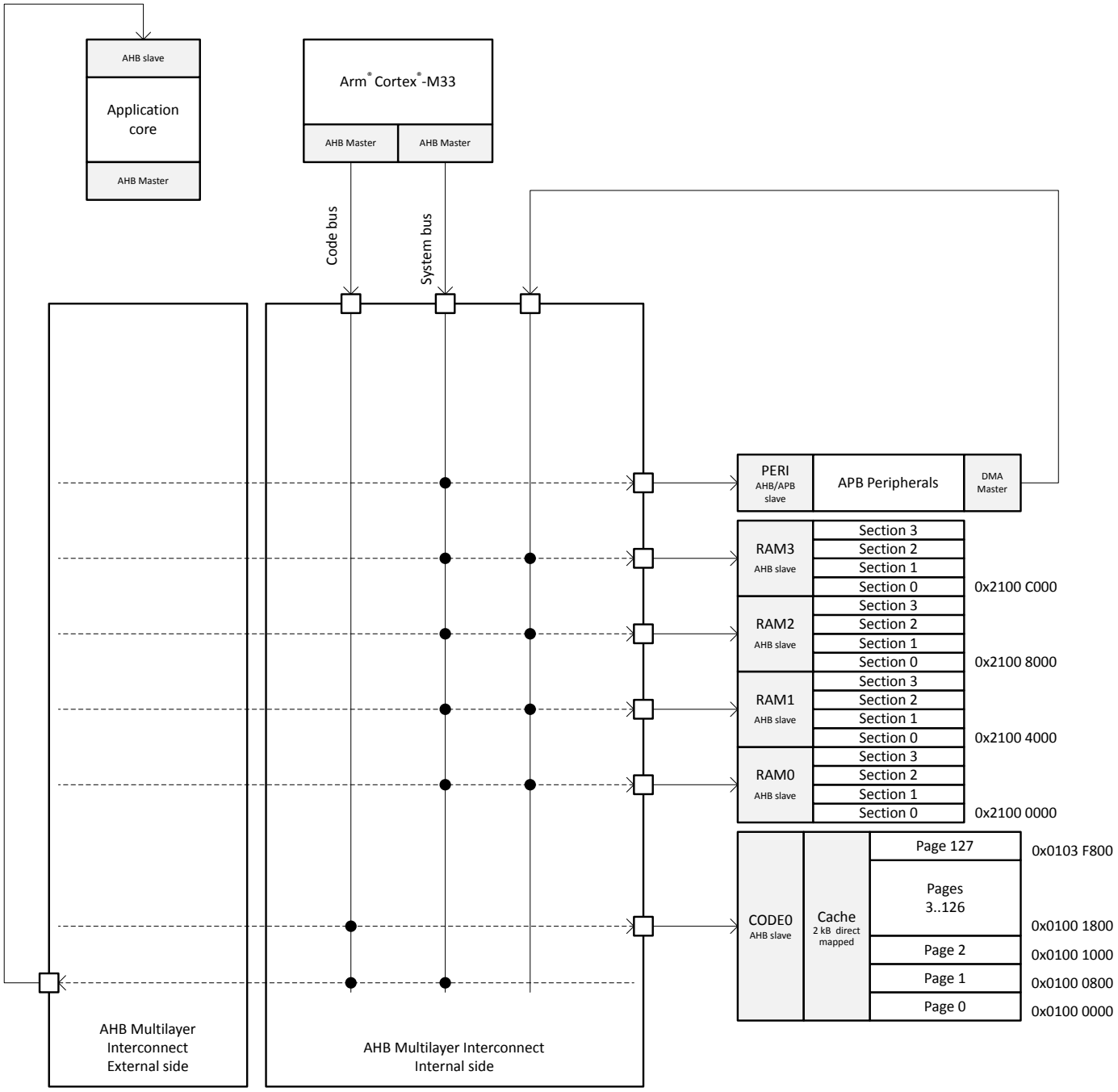

Figure 31: Memory layout

6.3.1 Peripheral instantiation

The following table describes the abbreviations used in the Instance, Secure mapping, and DMA security columns of the instantiation table.

Abbreviation	Description
NS	Non-secure - Peripheral is always accessible as a Non-Secure peripheral
S	Secure - Peripheral is always accessible as a Secure peripheral
US	User Selectable - A Secure or Non-secure attribute for the peripheral is defined in the SPU
SPLIT	Both Secure and Non-secure - The same resource is shared by both secure and non-secure code
NA	Not Applicable - Peripheral has no DMA capability
NSA	NoSeparateAttribute - Peripheral with DMA and DMA transfer has the same security attribute as assigned to the peripheral
SA	SeparateAttribute - Peripheral with DMA and DMA transfers can have a different security attribute than the one assigned to the peripheral

Table 43: Instantiation table abbreviations

The Secure mapping column in the following table defines configuration capabilities for the Arm TrustZone for Armv8-M secure attribute. The DMA security column describes the DMA capabilities of the peripheral.

6.3.2 Instantiation

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
0	0x41000000	DCNF	DCNF	NS	NA	Domain configuration
4	0x41004000	VREQCTRL	VREQCTRL	NS	NA	Voltage request control
5	0x41005000	CLOCK	CLOCK	NS	NA	Clock control
5	0x41005000	POWER	POWER	NS	NA	Power control
5	0x41005000	RESET	RESET	NS	NA	Reset status
6	0x41006000	CTRLAPPERI	CTRLAP	NS	NA	Control access port CPU side
8	0x41008000	RADIO	RADIO	NS	NA	2.4 GHz radio
9	0x41009000	RNG	RNG	NS	NA	Random number generator
10	0x4100A000	GPIOTE	GPIOTE	NS	NA	GPIO tasks and events
11	0x4100B000	WDT	WDT	NS	NA	Watchdog timer
12	0x4100C000	TIMER	TIMER0	NS	NA	Timer 0
13	0x4100D000	ECB	ECB	NS	NA	AES electronic code book (ECB) mode block encryption
14	0x4100E000	AAR	AAR	NS	NA	Accelerated address resolver
14	0x4100E000	CCM	CCM	NS	NA	AES counter with CBC-MAC (CCM) mode block encryption
15	0x4100F000	DPPIC	DPPIC	NS	NA	DPPI controller
16	0x41010000	TEMP	TEMP	NS	NA	Temperature sensor
17	0x41011000	RTC	RTC0	NS	NA	Real-time counter 0
18	0x41012000	IPC	IPC	NS	NA	Interprocessor communication
19	0x41013000	SPIM	SPIM0	NS	NA	SPI master 0
19	0x41013000	SPIS	SPIS0	NS	NA	SPI slave 0
19	0x41013000	TWIM	TWIM0	NS	NA	Two-wire interface master 0
19	0x41013000	TWIS	TWIS0	NS	NA	Two-wire interface slave 0
19	0x41013000	UARTE	UARTE0	NS	NA	Universal asynchronous receiver/transmitter
20	0x41014000	EGU	EGU0	NS	NA	Event generator unit 0
22	0x41016000	RTC	RTC1	NS	NA	Real-time counter 1
24	0x41018000	TIMER	TIMER1	NS	NA	Timer 1
25	0x41019000	TIMER	TIMER2	NS	NA	Timer 2

ID	Base address	Peripheral	Instance	Secure mapping	DMA security	Description
26	0x4101A000	SWI	SWI0	NS	NA	Software interrupt 0
27	0x4101B000	SWI	SWI1	NS	NA	Software interrupt 1
28	0x4101C000	SWI	SWI2	NS	NA	Software interrupt 2
29	0x4101D000	SWI	SWI3	NS	NA	Software interrupt 3
128	0x41080000	ACL	ACL	NS	NA	Access control lists
128	0x41080000	NVMC	NVMC	NS	NA	Non-Volatile Memory Controller
129	0x41081000	VMC	VMC	NS	NA	Volatile memory controller
192	0x418C0500	GPIO	P0	NS	NA	General purpose input and output
192	0x418C0800	GPIO	P1	NS	NA	General purpose input and output
N/A	0x01FF0000	FICR	FICR	NS	NA	Factory information configuration
N/A	0x01FF8000	UICR	UICR	NS	NA	User information configuration
N/A	0xE0042000	CTI	CTI	NS	NA	Cross-trigger interface

Table 44: Instantiation table

6.4 Core components

6.4.1 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.

6.4.1.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x01FF0000	NETWORK	FICR	FICR	NS	NA	Factory information configuration	

Table 45: Instances

Register	Offset	Security	Description
INFO.CONFIGID	0x200		Configuration identifier
INFO.DEVICEID[n]	0x204		Device identifier
INFO.PART	0x20C		Part code
INFO.VARIANT	0x210		Part Variant, Hardware version and Production configuration
INFO.PACKAGE	0x214		Package option
INFO.RAM	0x218		RAM variant
INFO.FLASH	0x21C		Flash variant
INFO.CODEPAGESIZE	0x220		Code memory page size in bytes
INFO.CODESIZE	0x224		Code memory size
INFO.DEVICETYPE	0x228		Device type
ER[n]	0x280		Encryption Root, word n
IR[n]	0x290		Identity Root, word n
DEVICEADDRTYPE	0x2A0		Device address type
DEVICEADDR[n]	0x2A4		Device address n
TRIMCNF[n].ADDR	0x300		Address
TRIMCNF[n].DATA	0x304		Data

Table 46: Register overview

6.4.1.1.1 INFO.CONFIGID

Address offset: 0x200

Configuration 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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	HWID			Identification number for the HW

6.4.1.1.2 INFO.DEVICEID[n] (n=0..1)

Address offset: 0x204 + (n × 0x4)

Device identifier

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

6.4.1.1.3 INFO.PART

Address offset: 0x20C

Part code

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00005340				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	PART			Part code
			N5340	0x5340	nRF5340
			Unspecified	0xFFFFFFFF	Unspecified

6.4.1.1.4 INFO.VARIANT

Address offset: 0x210

Part Variant, Hardware version and Production configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	VARIANT			Part Variant, Hardware version and Production configuration, encoded as ASCII
			QKAA	0x514B4141	QKAA
			Unspecified	0xFFFFFFFF	Unspecified

6.4.1.1.5 INFO.PACKAGE

Address offset: 0x214

Package option

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF					**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	PACKAGE			Package option
			QK	0x2000	QKxx - 94-pin aQFN
			Unspecified	0xFFFFFFFF	Unspecified

6.4.1.1.6 INFO.RAM

Address offset: 0x218

RAM variant

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF					**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	RAM			RAM variant
			K16	0x10	16 kByte RAM
			K32	0x20	32 kByte RAM
			K64	0x40	64 kByte RAM
			K128	0x80	128 kByte RAM
			K256	0x100	256 kByte RAM
			K512	0x200	512 kByte RAM
			Unspecified	0xFFFFFFFF	Unspecified

6.4.1.1.7 INFO.FLASH

Address offset: 0x21C

Flash variant

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

6.4.1.1.8 INFO.CODEPAGESIZE

Address offset: 0x220

Code memory page size in bytes

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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 0x00000800				**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 0**
ID	R/W	Field	Value ID	Value	Description
A	R	CODEPAGESIZE			Code memory page size in bytes
			K2048	0x800	2 kByte

6.4.1.1.9 INFO.CODESIZE

Address offset: 0x224

Code memory size

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000080				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	CODESIZE			Code memory size in number of pages
					Total code space is: CODEPAGESIZE * CODESIZE bytes
			P128	128	128 pages

6.4.1.1.10 INFO.DEVICETYPE

Address offset: 0x228

Device 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	DEVICETYPE			Device type
			Die	0x0000000	Device is an physical DIE
			FPGA	0xFFFFFFFF	Device is an FPGA

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

Address offset: 0x280 + (n × 0x4)

Encryption Root, word n

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

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

Address offset: 0x290 + (n × 0x4)

Identity Root, word n

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

6.4.1.1.13 DEVICEADDRTYPE

Address offset: 0x2A0

Device address type

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

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

Address offset: 0x2A4 + (n × 0x4)

Device address n

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

6.4.1.1.15 TRIMCNF[n].ADDR (n=0..31)

Address offset: 0x300 + (n × 0x8)

Address

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

6.4.1.1.16 TRIMCNF[n].DATA (n=0..31)

Address offset: 0x304 + (n × 0x8)

Data

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

6.4.2 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 registers, see NVMC — Non-volatile memory controller on page 329 and Memory on page 134.

6.4.2.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x01FF8000	NETWORK	UICR	UICR	NS	NA	User information configuration	

Table 47: Instances

Register	Offset	Security	Description
APPROTECT	0x000		Access port protection
ERASEPROTECT	0x004		Erase protection
NRFFW[n]	0x200		Reserved for Nordic firmware design
CUSTOMER[n]	0x300		Reserved for customer

Table 48: Register overview

6.4.2.1.1 APPROTECT

Address offset: 0x000

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	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	R/W	Field	Value ID	Value																															
A	RW	PALL																																	
				Blocks debugger read/write access to all CPU registers and memory mapped addresses. Using any value except Unprotected will lead to the protection being enabled. Bits with value '1' can be written to '0'. Bits with value '0' cannot be written to '1'.																																
			Unprotected	0x50FA50FA																																	
			Protected	0x00000000																																	
				Protected																																	

6.4.2.1.2 ERASEPROTECT

Address offset: 0x004

Erase 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 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	R/W	Field	Value ID	Value	Description
A	RW	PALL			Blocks NVMC ERASEALL and CTRLAP ERASEALL functionality. Using any value except Unprotected will lead to the protection being enabled.
			Unprotected	0xFFFFFFFF	Unprotected
			Protected	0x00000000	Protected

6.4.2.1.3 NRFFW[n] (n=0..31)

Address offset: 0x200 + (n × 0x4)

Reserved for Nordic firmware design

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

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

Address offset: 0x300 + (n × 0x4)

Reserved for customer

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

6.4.3 AHB multilayer

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

Each bus master is connected to a slave device through one or more interconnection matrixes. The bus masters are assigned priorities that are used to resolve access when two (or more) bus masters request access to the same slave device. The following applies when assigning priorities:

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

Some peripherals, like RADIO, do not have a safe stalling mechanism (not able to pause incoming data and no internal data buffering). Being a low priority bus master might cause loss of data for such peripherals

upon bus contention. To avoid AHB bus contention when using multiple bus masters, apply one of the following guidelines:

- Avoid situations where more than one bus master is accessing the same slave.
- If more than one bus master is accessing the same slave, make sure that the bus bandwidth is not exhausted.

6.4.3.1 AHB multilayer priorities

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

Bus master name	Natural relative priority	In/Out	Description
CPU	Highest priority	I/O	
RADIO		I/O	
CCM/ECB/AAR		I/O	Same priority and mutually exclusive
UARTE0/SPIM0/SPIS0/TWIM0/TWIS0	Lowest priority	I/O	Same priority and mutually exclusive

Table 49: AHB bus masters

7 Peripherals

nRF5340 is made up of two cores, each with theis own peripherals.

The application core peripherals are found in Instantiation on page 105 and the network core peripherals are found in Instantiation on page 136. The application core peripherals are accessible from the network core, but the network core peripherals are not accessible from the application core. For further details, see Memory on page 18.

Some peripheral instances have differing configurations, such as some TIMER instances which have more capture and compare channels implemented than others. These differences are noted in the Configuration column of the peripheral's instantiation table.

7.1 Peripheral interface

Peripherals are controlled by the CPU through configuration registers, as well as task and event registers. Task registers are inputs, enabling the CPU and other peripherals to initiate a functionality. Event registers are outputs, enabling a peripheral to trigger tasks in other peripherals and/or the CPU by tying events to CPU interrupts.

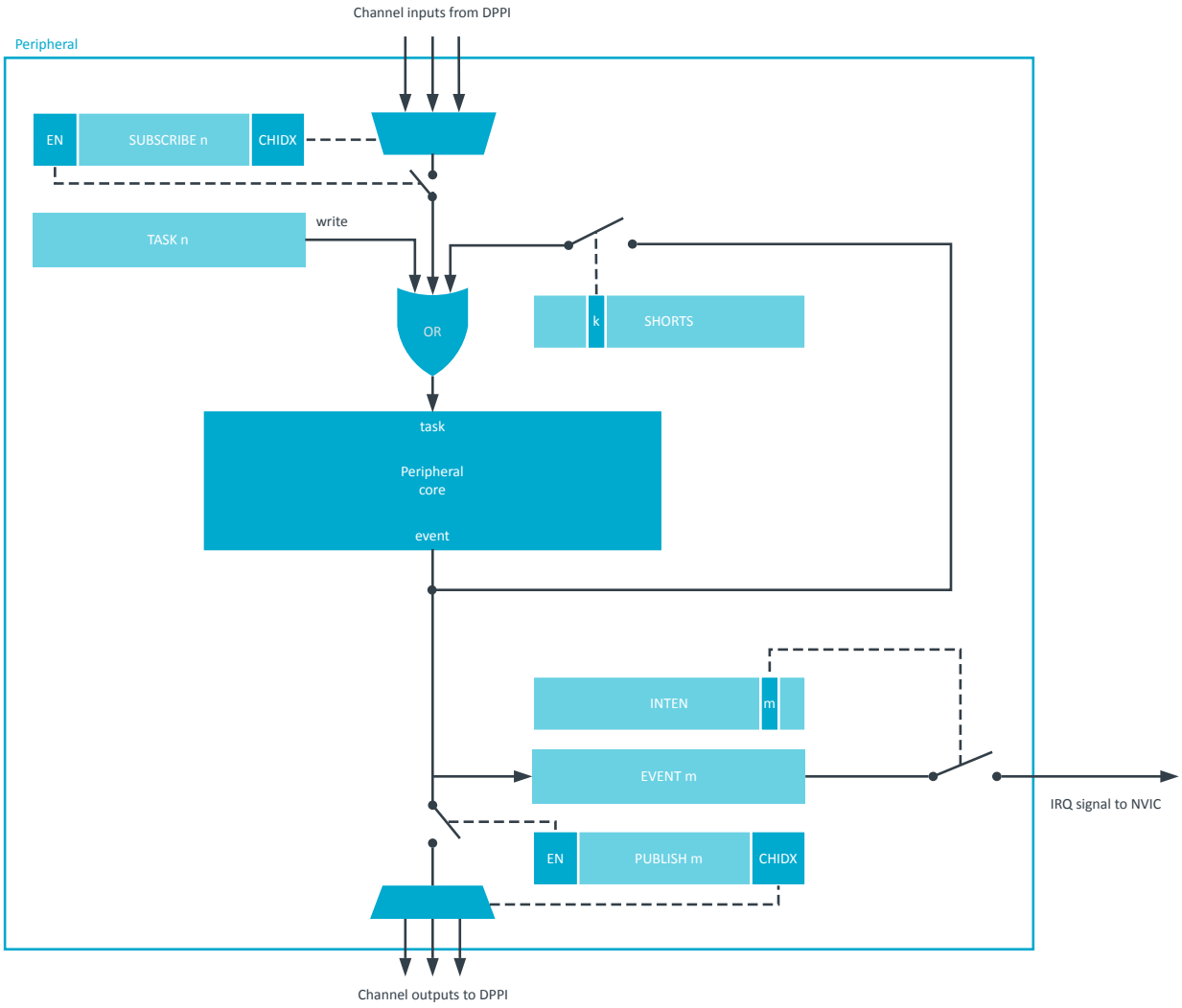

Figure 32: Peripheral interface

The distributed programmable peripheral interconnect (DPPI) feature enables peripherals to connect events to tasks without CPU intervention. For more information on DPPI and the DPPI channels, see DPPI - Distributed programmable peripheral interconnect on page 198.

7.1.1 Peripheral ID

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

See Peripherals on page 145 for more information about which peripherals are available and where they are located in the address map.

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

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

- Shared registers or common resources
- Limited availability due to mutually exclusive operation; only one peripheral in use at a time
- Enforced peripheral behavior when switching between peripherals (disable the first peripheral before enabling the second)

7.1.2 Peripherals with shared ID

In general (with the exception of ID 0), peripherals sharing an ID and base address may not be used simultaneously. Only one peripheral can be enabled at a given ID.

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

1. Disable the previously used peripheral.
2. Disable any publish/subscribe connection to the DPPI system for the peripheral that is being disabled.
3. Clear all bits in the INTEN register, i.e. INTENCLR = 0xFFFFFFFF.
4. Explicitly configure the peripheral being enabled. Do not rely on inherited configuration from the disabled peripheral.
5. Enable the now configured peripheral.

For a list of which peripherals that share an ID see Peripherals on page 145.

7.1.3 Peripheral registers

Most peripherals feature an ENABLE register. Unless otherwise specified, the peripheral registers must be configured before enabling the peripheral.

PSEL registers need to be set before a peripheral is enabled or started. Updating PSEL registers while the peripheral is running has no effect. To connect a peripheral to a different GPIO, the following must be performed:

1. Disable the peripheral.
2. Update the PSEL register.
3. Re-enable the peripheral.

It takes four CPU cycles between the PSEL register update and the connection between a peripheral and a GPIO becoming effective.

Note: The peripheral must be enabled before tasks and events can be used.

Most of the register values are lost during System OFF or when a reset is triggered. Some registers will retain their values in System OFF or for some specific reset sources. These registers are marked as retained in the register description for a given peripheral. For more information on their behavior, see chapter RESET — Reset control on page 61.

7.1.4 Bit set and clear

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

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

In the main register, the SET register sets individual bits and the CLR register clears them. Writing `1` to a bit in the SET or CLR register will set or clear the same bit in the main register respectively. Writing `0` to a bit in the SET or CLR register has no effect. Reading the SET or CLR register returns the value of the main register.

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

7.1.5 Tasks

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

A task is triggered when firmware writes `1` to the task register, or when the peripheral itself or another peripheral toggles the corresponding task signal. See the figure Peripheral interface on page 146.

7.1.6 Events

Events are used to notify peripherals and the CPU about events that have happened, for example a state change in a peripheral. A peripheral may generate multiple events, where each event has 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 Peripheral interface on page 146. An event register is cleared when a `0` is written to it by firmware. Events can be generated by the peripheral even when the event register is set to `1`.

7.1.7 Publish and subscribe

Events and tasks from different peripherals can be connected together through the DPPI system using the PUBLISH and SUBSCRIBE registers in each peripheral. See Peripheral interface on page 146. An event can be published onto a DPPI channel by configuring the event's PUBLISH register. Similarly, a task can subscribe to a DPPI channel by configuring the task's SUBSCRIBE register.

See DPPI - Distributed programmable peripheral interconnect on page 198 for details.

7.1.8 Shortcuts

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

Using shortcuts is equivalent to making the connection outside the peripheral and through the DPPI. However, the propagation delay when using shortcuts is usually shorter than the propagation delay through the DPPI.

Shortcuts are predefined, which means that 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.

7.1.9 Interrupts

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

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

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

The relationship between tasks, events, shortcuts, and interrupts is illustrated in figure Peripheral interface on page 146.

7.1.9.1 Interrupt clearing and disabling

Interrupts should always be cleared by writing `0` to the corresponding EVENT register.

Until cleared, interrupts will immediately be re-triggered and cause software interrupt service routines to be executed repeatedly.

Because the clearing of the EVENT register may take a number of CPU clock cycles, the program should perform a read from the EVENT register that has been cleared before exiting the interrupt service routine. This will ensure that the EVENT clearing has taken place before the interrupt service routine is exited. Care should be taken to ensure that the compiler does not remove the read operation as an optimization.

Similarly, when disabling an interrupt inside an interrupt service routine, the program should perform a read from the INTEN or INTENCLR registers to ensure that the interrupt is disabled before exiting the interrupt service routine.

7.1.10 Secure/non-secure peripherals

For some peripherals, the security configuration can change from secure to non-secure, or vice versa. Care must be taken when changing the security configuration of a peripheral, to prevent security information leakage and ensure correct operation.

The following sequence should be followed, where applicable, when configuring and changing the security settings of a peripheral in the SPU — System protection unit on page 584.

1. Stop peripheral operation.
2. Disable the peripheral.
3. Remove pin connections.
4. Disable DPPI connections.
5. Clear sensitive registers (e.g. writing back default values).
6. Change peripheral security setting in the SPU — System protection unit on page 584.
7. Re-enable the peripheral.

7.2 EasyDMA

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

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

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

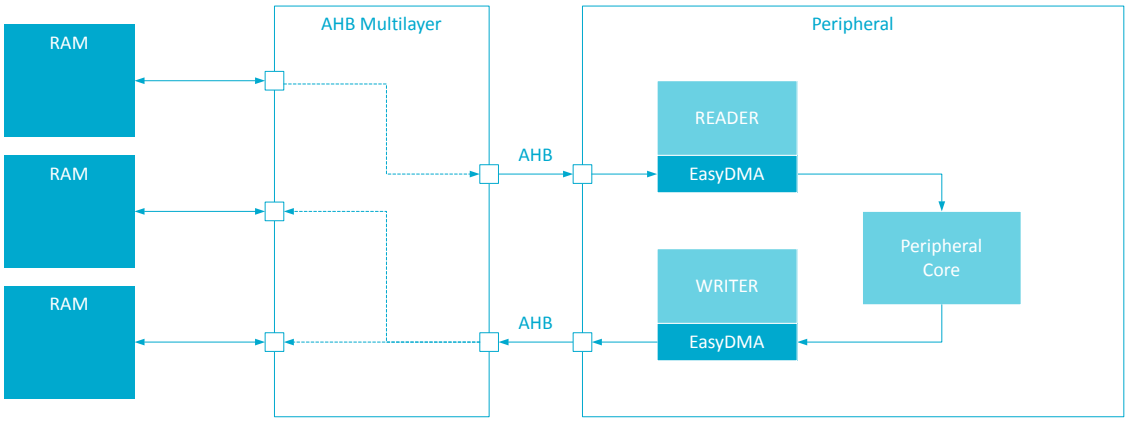

Figure 33: EasyDMA example

An EasyDMA channel is implemented in the following way, but some variations may occur:

```
READERBUFFER_SIZE 5
WRITERBUFFER_SIZE 6

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

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

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

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

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

The memory layout of these buffers is illustrated in EasyDMA memory layout on page 150.

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 34: EasyDMA memory layout

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

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

Note: The PTR register of a READER or WRITER must point to a valid memory region before use. The reset value of a PTR register is not guaranteed to point to valid memory. See Memory on page 18 for more information about the different memory regions and EasyDMA connectivity.

7.2.1 EasyDMA error handling

Some errors may occur during DMA handling.

If READER.PTR or WRITER.PTR is not pointing to a valid memory region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 18 for more information about the different memory regions.

If several AHB bus masters try to access the same AHB slave at the same time, AHB bus congestion might occur. An EasyDMA channel is an AHB master. Depending on the peripheral, the peripheral may either stall and wait for access to be granted, or lose data.

7.2.2 EasyDMA array list

EasyDMA is able to operate in Array List mode.

The Array List mode is implemented in channels where the LIST register is available.

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

The EasyDMA Array List can be implemented by using the data structure ArrayList_type as illustrated in the code example below using a READER EasyDMA channel as an example:

```
#define BUFFER_SIZE  4

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

ArrayList_type ReaderList[3]  __at__ 0x20000000;

MYPERIPHERAL->READER.MAXCNT = BUFFER_SIZE;
MYPERIPHERAL->READER.PTR = &ReaderList;
MYPERIPHERAL->READER.LIST = MYPERIPHERAL_READER_LIST_ArrayList;
```

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

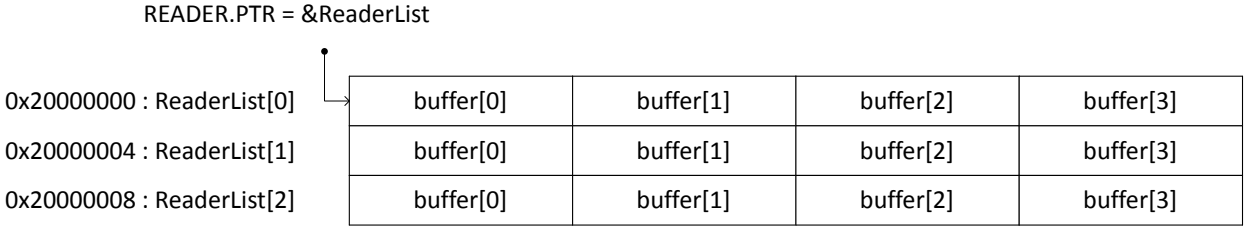

Figure 35: EasyDMA array list

7.3 ACL — Access control lists

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

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

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

Note: The size of the region in bytes is restricted to a multiple of the flash page size, and the maximum region size is limited to the flash size. See Memory on page 18 for more information.

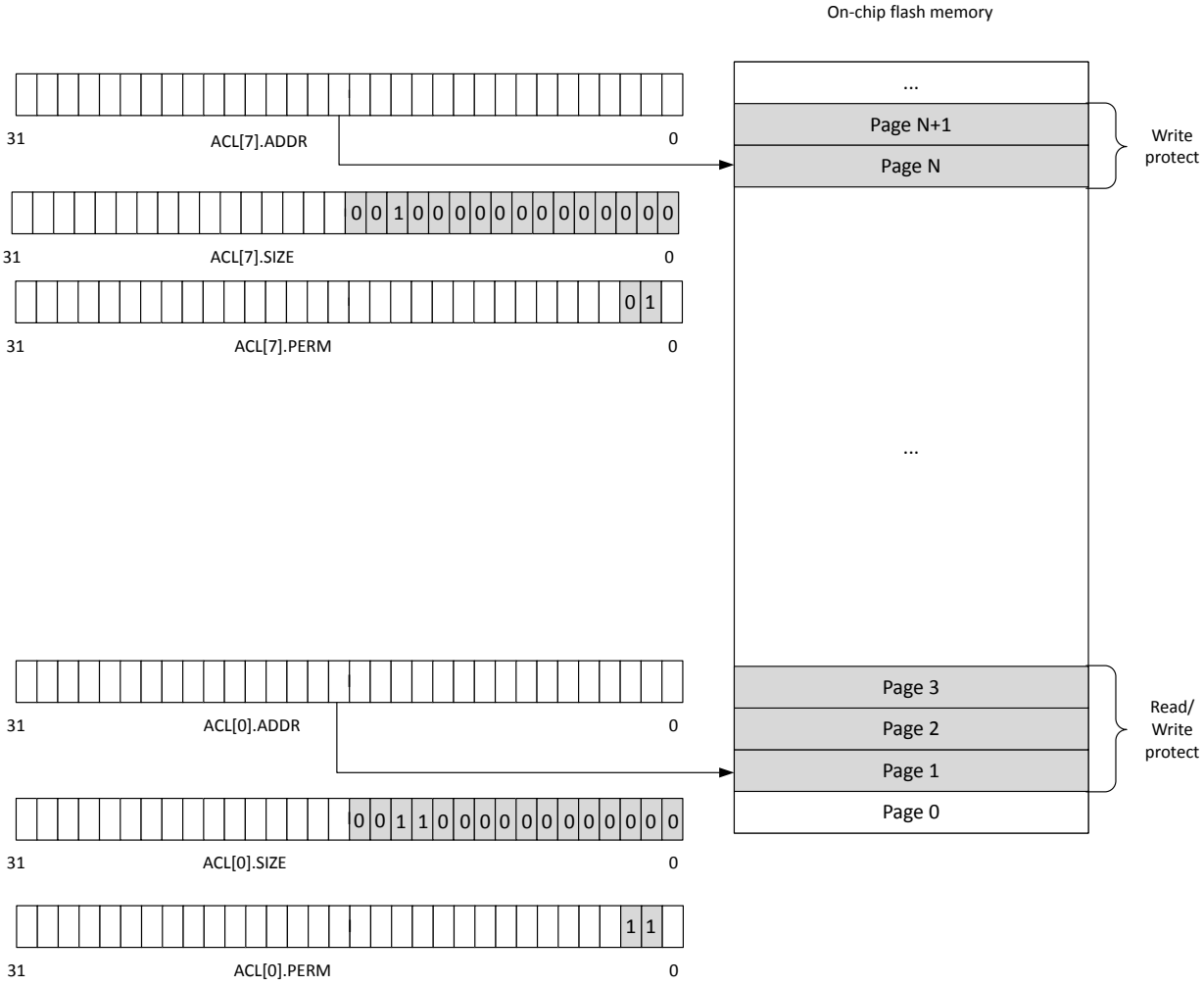

Figure 36: Protected regions of on-chip flash memory

There are four defined ACL permission schemes, each with different combinations of read/write permissions, as shown in the following table.

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

Table 50: Permission schemes

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

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

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

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

7.3.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x41080000	NETWORK	ACL	ACL	NS	NA	Access control lists	This ACL can only protect network core's local memory.

Table 51: Instances

Register	Offset	Security	Description
ACL[n].ADDR	0x800		Start address of region to protect. The start address must be word-aligned.
ACL[n].SIZE	0x804		Size of region to protect counting from address ACL[n].ADDR. Writing a '0' has no effect.
ACL[n].PERM	0x808		Access permissions for region n as defined by start address ACL[n].ADDR and size ACL[n].SIZE
ACL[n].UNUSED0	0x80C		Reserved

Table 52: Register overview

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

Address offset: 0x800 + (n × 0x10)

Start address of region to protect. The start address must be word-aligned.

Note: This register can only be written once.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW1	ADDR			Start address of flash region n. The start address must point to a flash page boundary.

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

Address offset: 0x804 + (n × 0x10)

Size of region to protect counting from address ACL[n].ADDR. Writing a '0' has no effect.

Note: This register can only be written once.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW1	SIZE			Size of flash region n in bytes. Must be a multiple of the flash page size.

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

Address offset: 0x808 + (n × 0x10)

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

Note: This register can only be written once.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
B	RW1	WRITE			Configure write and erase permissions for region n. Writing a '0' has no effect.
			Enable	0	Allow write and erase instructions to region n.
			Disable	1	Block write and erase instructions to region n.
C	RW1	READ			Configure read permissions for region n. Writing a '0' has no effect.
			Enable	0	Allow read instructions to region n.
			Disable	1	Block read instructions to region n.

7.4 AAR — Accelerated address resolver

Accelerated address resolver is a cryptographic support function for implementing the Resolvable Private Address Resolution procedure described in *Bluetooth Core Specification*. 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.

7.4.1 Shared resources

AAR shares the same AES module as the ECB and CCM peripherals. ECB will always have the lowest priority. If there is a sharing conflict during encryption, the ECB operation will be aborted and an ERRORECB event will be generated in the ECB peripheral.

Additionally, AAR shares registers and other resources with the peripherals that have the same ID as AAR. See Peripherals with shared ID on page 147 for more information.

7.4.2 EasyDMA

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

If the IRKPTR on page 161, ADDRPTR on page 161, and the SCRATCHPTR on page 161 is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 18 for more information about the different memory regions.

7.4.3 Resolving a resolvable address

A private resolvable address is composed of six bytes according to the *Bluetooth Core Specification*.

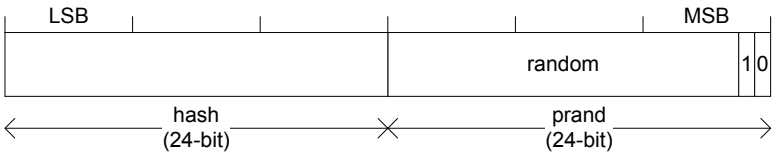

Figure 37: Resolvable address

To resolve an address, the register ADDRPTR on page 161 must point to the start of the packet. The resolver is started by triggering the START task. A RESOLVED event is generated when AAR manages to resolve the address using one of the Identity Resolving Keys (IRK) found in the IRK data structure. AAR will use the IRK specified in the register IRK0 to IRK15 starting from IRK0. The register NIRK on page 161 specifies how many IRKs should be used. The AAR module will generate a NOTRESOLVED event if it is not able to resolve the address using the specified list of IRKs.

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 Core specification*. The time it takes to resolve an address varies due to the location in the list of the resolvable address. 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.

AAR only compares the received address to those programmed in the module without checking the address type.

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. AAR will generate an END event after it has stopped.

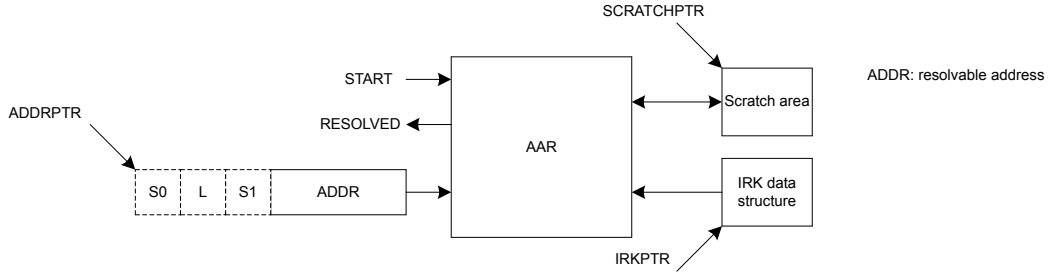

Figure 38: Address resolution with packet preloaded into RAM

7.4.4 Example

The following example shows how to chain RADIO packet reception with address resolution using AAR.

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

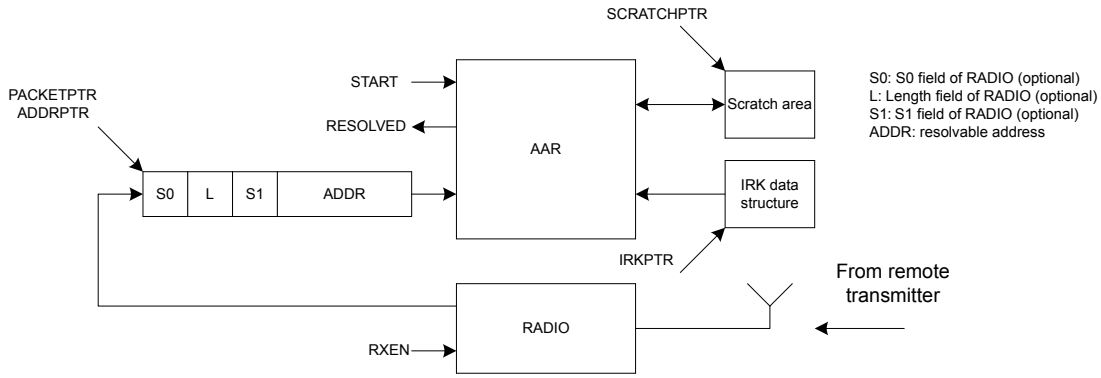

Figure 39: Address resolution with packet loaded into RAM by RADIO

7.4.5 IRK data structure

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

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

Table 53: IRK data structure overview

7.4.6 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x4100E000	NETWORK	AAR	AAR	NS	NA	Accelerated address resolver	

Table 54: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start resolving addresses based on IRKs specified in the IRK data structure
TASKS_STOP	0x008		Stop resolving addresses
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x088		Subscribe configuration for task STOP
EVENTS_END	0x100		Address resolution procedure complete
EVENTS_RESOLVED	0x104		Address resolved
EVENTS_NOTRESOLVED	0x108		Address not resolved
PUBLISH_END	0x180		Publish configuration for event END
PUBLISH_RESOLVED	0x184		Publish configuration for event RESOLVED
PUBLISH_NOTRESOLVED	0x188		Publish configuration for event NOTRESOLVED
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
STATUS	0x400		Resolution status

Register	Offset	Security	Description
ENABLE	0x500		Enable AAR
NIRK	0x504		Number of IRKs
IRKPTR	0x508		Pointer to IRK data structure
ADDRPTR	0x510		Pointer to the resolvable address
SCRATCHPTR	0x514		Pointer to data area used for temporary storage

Table 55: Register overview

7.4.6.1 TASKS_START

Address offset: 0x000

Start resolving addresses based on IRKs specified in the 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start resolving addresses based on IRKs specified in the IRK data structure
			Trigger	1	Trigger task

7.4.6.2 TASKS_STOP

Address offset: 0x008

Stop resolving addresses

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop resolving addresses
			Trigger	1	Trigger task

7.4.6.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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	B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.4.6.4 SUBSCRIBE_STOP

Address offset: 0x088

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.4.6.5 EVENTS_END

Address offset: 0x100

Address resolution procedure complete

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			Address resolution procedure complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.4.6.6 EVENTS_RESOLVED

Address offset: 0x104

Address resolved

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RESOLVED			Address resolved
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.4.6.7 EVENTS_NOTRESOLVED

Address offset: 0x108

Address not resolved

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_NOTRESOLVED			Address not resolved
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.4.6.8 PUBLISH_END

Address offset: 0x180

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.4.6.9 PUBLISH_RESOLVED

Address offset: 0x184

Publish configuration for event RESOLVED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RESOLVED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.4.6.10 PUBLISH_NOTRESOLVED

Address offset: 0x188

Publish configuration for event NOTRESOLVED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event NOTRESOLVED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.4.6.11 INTENSET

Address offset: 0x304

Enable interrupt

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

7.4.6.12 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	RESOLVED			Write '1' to disable interrupt for event RESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	NOTRESOLVED			Write '1' to disable interrupt for event NOTRESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.4.6.13 STATUS

Address offset: 0x400

Resolution status

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

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

7.4.6.14 ENABLE

Address offset: 0x500

Enable AAR

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

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

7.4.6.15 NIRK

Address offset: 0x504

Number of IRKs

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

ID	R/W	Field	Value ID	Value	Description
A	RW	NIRK		[1..16]	Number of Identity Root Keys available in the IRK data structure

7.4.6.16 IRKPTR

Address offset: 0x508

Pointer to IRK data structure

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

ID	R/W	Field	Value ID	Value	Description
A	RW	IRKPTR			Pointer to the IRK data structure

7.4.6.17 ADDRPTR

Address offset: 0x510

Pointer to the resolvable address

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ADDRPTR			Pointer to the resolvable address (6-bytes)

7.4.6.18 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SCRATCHPTR			Pointer to a scratch data area used for temporary storage during resolution. A space of minimum 3 bytes must be reserved.

7.4.7 Electrical specification

7.4.7.1 AAR Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{AAR}	Address resolution time per IRK. Total time for several IRKs is given as (1 µs + n * t_AAR), where n is the number of IRKs. (Given priority to the actual destination RAM block).			6.1	µs
$t_{AAR,8}$	Time for address resolution of 8 IRKs. (Given priority to the actual destination RAM block).			49	µs

7.5 CCM — AES CCM mode encryption

Counter with cipher block chaining - message authentication code (CCM) mode is an authenticated encryption algorithm designed to provide both authentication and confidentiality during data transfer.

AES CCM combines counter (CTR) mode encryption and cipher block chaining - message authentication code (CBC-MAC) authentication. The CCM terminology message authentication code (MAC) is called 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 four byte MIC field in one operation. CCM and RADIO can be configured to work synchronously. CCM will encrypt in time for transmission and decrypt after receiving bytes into memory from 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 Bluetooth Low Energy.

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

Three operations are supported:

- Keystream generation
- Packet encryption
- Packet decryption

All operations are done in compliance with the *Bluetooth* specification.[8]

The following figure illustrates keystream generation followed by encryption or decryption. The shortcut is optional.

[8] *Bluetooth* AES CCM 128-bit block encryption, see *Bluetooth Core specification* version 4.0.

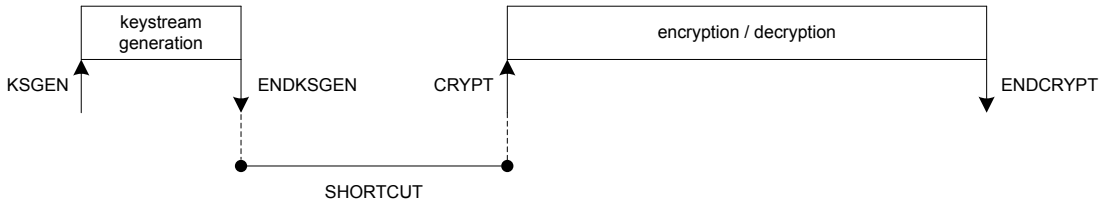

Figure 40: Keystream generation

7.5.1 Shared resources

CCM shares the same AES module as the ECB and AAR peripherals. ECB will always have the lowest priority. If there is a sharing conflict during encryption, the ECB operation will be aborted and an ERRORECB event will be generated in ECB.

Additionally, CCM shares registers and other resources with other peripherals that have the same ID as CCM. See Peripherals with shared ID on page 147 for more information.

7.5.2 Keystream generation

A new keystream needs to be generated before a new packet encryption or packet decryption operation can start.

A keystream is generated by triggering the KSGEN task. An ENDKSGEN event is generated after the keystream has been generated.

Keystream generation, packet encryption, and packet decryption operations utilize the configuration specified in the data structure pointed to by CNFPTR on page 175. It is necessary to configure this pointer and its underlying data structure, and register MODE on page 174 before the KSGEN task is triggered.

The keystream will be stored in CCM's temporary memory area, specified by the SCRATCHPTR on page 176, where it will be used in subsequent encryption and decryption operations.

For default length packets (MODE.LENGTH = Default), the size of the generated keystream is 27 bytes. When using extended length packets (MODE.LENGTH = Extended), register MAXPACKETSIZE on page 176 specifies the length of the keystream to be generated. The length of the generated keystream must be greater than or equal to the length of the subsequent packet payload to be encrypted or decrypted. The maximum length of the keystream in extended mode is 251 bytes, which means that the maximum packet payload size is 251 bytes.

If a shortcut is used between ENDKSGEN event and CRYPT task, pointers INPTR on page 175 and OUTPTR on page 175 must also be configured before the KSGEN task is triggered.

7.5.3 Encryption

CCM is able to read an unencrypted packet, encrypt it, and append a four byte MIC field to the packet.

During packet encryption, CCM performs the following:

- Reads the unencrypted packet located in RAM address specified in the INPTR pointer
- Encrypts the packet
- Appends a four byte long Message Integrity Check (MIC) field to the packet

Encryption is started by triggering the CRYPT task with register MODE on page 174 set to `Encryption`. An ENDCRYPT event is generated when packet encryption is completed.

CCM will also modify the length field of the packet to adjust for the appended MIC field. It adds four bytes to the length and stores the resulting packet in RAM at the address specified in pointer OUTPTR on page 175, as illustrated in the figure below.

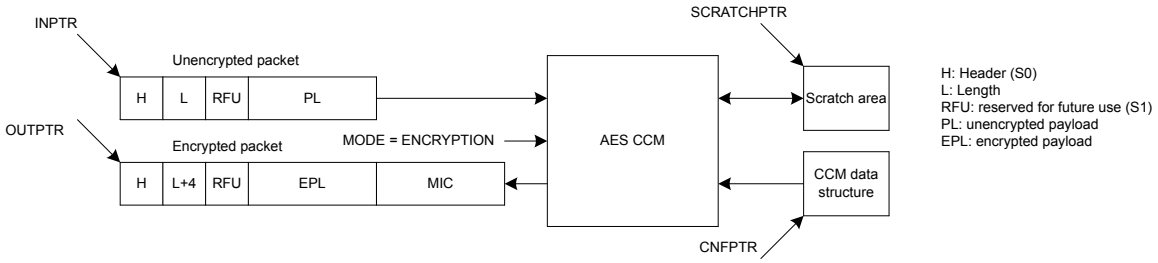

Figure 41: Encryption

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

CCM supports different widths of the length field in the data structure for encrypted packets. This is configured in register MODE on page 174.

7.5.4 Decryption

CCM is able to read an encrypted packet, decrypt it, authenticate the MIC field, and generate an appropriate MIC status.

During packet decryption, CCM performs the following:

- Reads the encrypted packet located in RAM at the address specified in the INPTR pointer
- Decrypts the packet
- Authenticates the packet's MIC field
- Generates the appropriate MIC status

The packet header (S0) and payload are included in the MIC authentication. Bits in the packet header can be masked away by configuring register HEADERMASK on page 177.

Decryption is started by triggering the CRYPT task, by setting the register MODE on page 174 to `Decryption`. An ENDCRYPT event will be generated when packet decryption is completed.

CCM modifies the length field of the packet to adjust for the MIC field. It subtracts four bytes from the length and stores the decrypted packet in RAM at the address specified in the OUTPTR on page 175 pointer.

CCM is only able to decrypt packet payloads that are at least five bytes long (one byte or more encrypted payload (EPL) and four bytes of MIC). 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 CCM. These packets will always pass the MIC check.

CCM supports different widths of the LENGTH field in the data structure for decrypted packets. This is configured in register MODE on page 174.

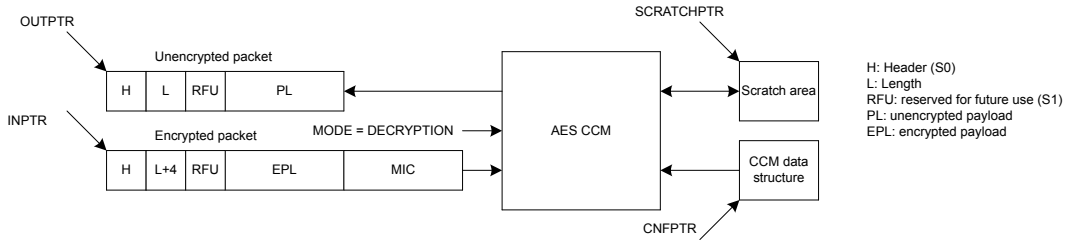

Figure 42: Decryption

The CCM is only able to decrypt packet payloads that are at least 5 bytes long, 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 CCM. These packets will always pass the MIC check.

CCM supports different widths of the length field in the data structure for decrypted packets. This is configured in register MODE on page 174.

7.5.5 AES CCM and RADIO concurrent operation

The CCM peripheral is able to encrypt/decrypt data synchronously to data being transmitted or received by RADIO.

In order for CCM to run synchronously with the radio, the data rate setting in register MODE on page 174 needs to match the radio data rate. The settings in this register apply whenever either the KSGEN or CRYPT tasks are triggered.

The data rate setting of register MODE on page 174 can also be overridden on-the-fly during an ongoing encrypt/decrypt operation by the contents of register RATEOVERRIDE on page 176. The data rate setting in this register applies whenever the RATEOVERRIDE task is triggered. This feature can be useful in cases where the radio data rate is changed during an ongoing packet transaction.

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

When CCM encrypts a packet on-the-fly while RADIO is transmitting it, RADIO must read the encrypted packet from the same memory location that CCM is writing to.

The OUTPTR on page 175 pointer in CCM must point to the same memory location as the PACKETPTR pointer in RADIO, as illustrated in the following figure.

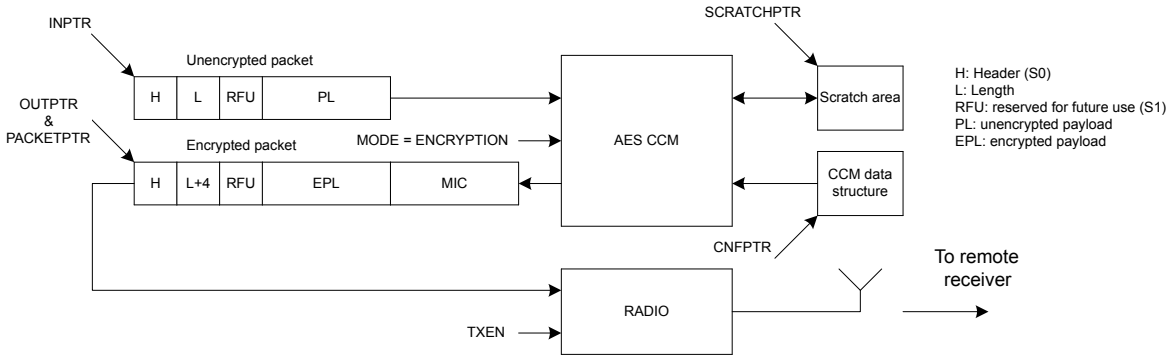

Figure 43: Configuration of on-the-fly encryption

In order to match RADIO's timing, the KSGEN task must be triggered early enough to allow the keystream generation to complete before packet encryption begins.

For short packets (MODE.LENGTH = Default), the KSGEN task must be triggered before or at the same time as the START task in RADIO is triggered. In addition, the shortcut between the ENDKSGEN event and the CRYPT task must be enabled. This use-case is illustrated in On-the-fly encryption of short packets (MODE.LENGTH = Default), using a PPI connection on page 166. It uses a PPI connection between the READY event in RADIO and the KSGEN task in CCM.

For long packets (MODE.LENGTH = Extended), the keystream generation needs to start earlier, such as when the TXEN task in RADIO is triggered.

Refer to Timing specification on page 177 for information about the time needed for generating a keystream.

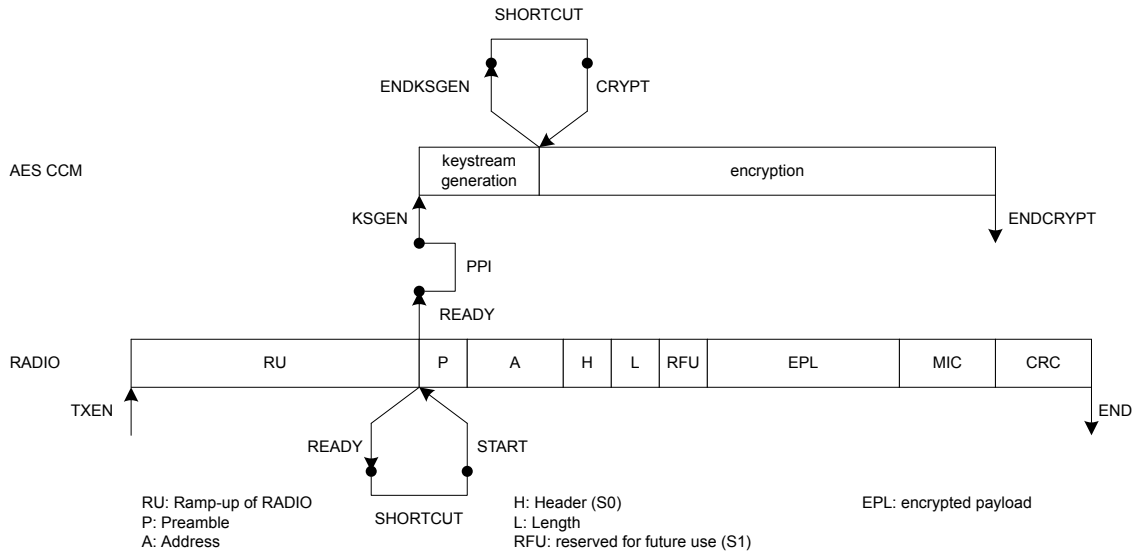

Figure 44: On-the-fly encryption of short packets (MODE.LENGTH = Default), using a PPI connection

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

Note: See Timing specification on page 177 for information about the time needed for generating a keystream.

7.5.7 Decrypting packets on-the-fly in RADIO receive mode

When CCM decrypts a packet on-the-fly while RADIO is receiving it, CCM must read the encrypted packet from the same memory location that RADIO is writing to.

The pointer INPTR on page 175 in CCM must therefore point to the same memory location as the PACKETPTR pointer in RADIO.

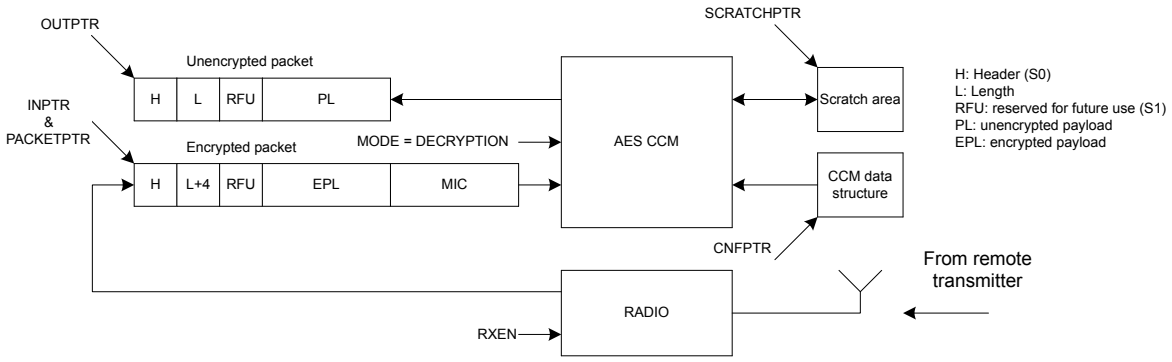

Figure 45: Configuration of on-the-fly decryption

In order to match RADIO's timing, the KSGEN task must be triggered early enough to allow the keystream generation to complete before the decryption of the packet starts.

For short packets (MODE.LENGTH = Default), the KSGEN task must be triggered no later than when the START task in RADIO is triggered. In addition, the CRYPT task must not be triggered earlier than when the ADDRESS event is generated by RADIO. If the CRYPT task is triggered exactly at the same time as the ADDRESS event is generated by RADIO, CCM will guarantee that the decryption is completed no later than when the END event in RADIO is generated. This use case, using a PPI connection between the ADDRESS event in RADIO and the CRYPT task in CCM, is illustrated in figure below.

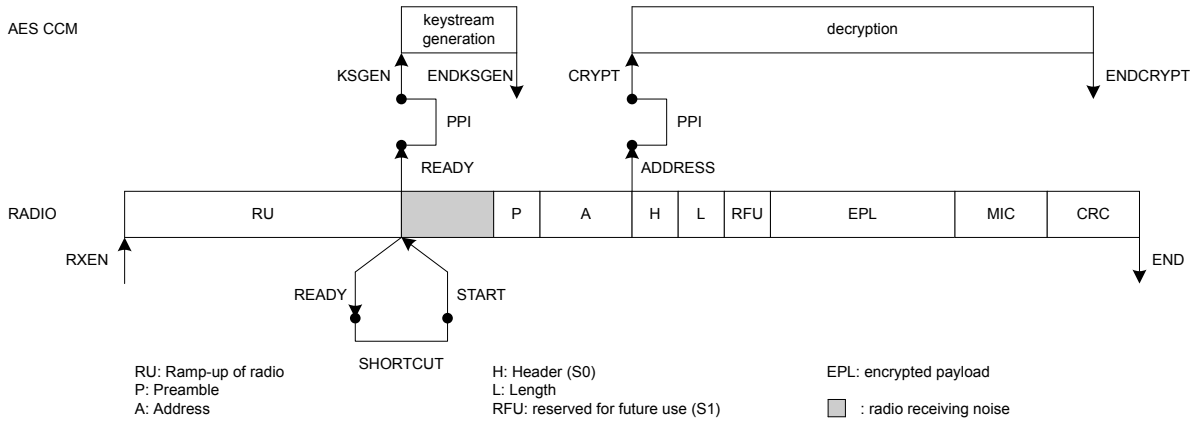

Figure 46: On-the-fly decryption of short packets (MODE.LENGTH = Default), using a PPI connection

The KSGEN task is triggered from the READY event in RADIO, through a PPI connection.

For long packets (MODE.LENGTH = Extended) the keystream generation will need to start even earlier, such as when the RXEN task in RADIO is triggered.

Refer to Timing specification on page 177 for information about the time needed for generating a keystream.

7.5.8 CCM data structure

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

Property	Address offset	Description
KEY	0	16-byte AES key.
PKTCTR	16	Octet0 (least significant octet (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). Bit 7: Ignored.
	21	Ignored.
	22	Ignored.
	23	Ignored.
	24	Bit 0: Direction bit. Bit 7 – bit 1: Zero padded.
IV	25	8-byte initialization vector (IV). Octet0 (LSO) of IV, Octet1 of IV, … , Octet7 (MSO) of IV.

Table 56: CCM data structure overview

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

Property	Address offset	Description
HEADER	0	Packet header
LENGTH	1	Number of bytes in unencrypted payload
RFU	2	Reserved for future use
PAYLOAD	3	Unencrypted payload

Table 57: Data structure for unencrypted packet

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

Table 58: Data structure for encrypted packet

7.5.9 EasyDMA and ERROR event

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

When the CPU and EasyDMA enabled peripherals access the same RAM block at the same time, increased bus collisions might disrupt on-the-fly encryption. In this case, the ERROR event will be generated.

EasyDMA stops accessing RAM when the ENDKSGEN and ENDCRYPT events are generated.

If the CNFPTR, SCRATCHPTR, INPTR and the OUTPTR are not pointing to the data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 18 for more information about the different memory regions.

7.5.10 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x4100E000	NETWORK	CCM	CCM	NS	NA	AES counter with CBC-MAC (CCM) mode block encryption	

Table 59: Instances

Register	Offset	Security	Description	
TASKS_KSGEN	0x000		Start generation of keystream. This operation will stop by itself when completed.	
TASKS_CRYPT	0x004		Start encryption/decryption. This operation will stop by itself when completed.	
TASKS_STOP	0x008		Stop encryption/decryption	
TASKS_RATEOVERRIDE	0x00C		Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption	
SUBSCRIBE_KSGEN	0x080		Subscribe configuration for task KSGEN	
SUBSCRIBE_CRYPT	0x084		Subscribe configuration for task CRYPT	
SUBSCRIBE_STOP	0x088		Subscribe configuration for task STOP	
SUBSCRIBE_RATEOVERRIDE	0x08C		Subscribe configuration for task RATEOVERRIDE	
EVENTS_ENDKSGEN	0x100		Keystream generation complete	
EVENTS_ENDCRYPT	0x104		Encrypt/decrypt complete	
EVENTS_ERROR	0x108		CCM error event	Deprecated
PUBLISH_ENDKSGEN	0x180		Publish configuration for event ENDKSGEN	
PUBLISH_ENDCRYPT	0x184		Publish configuration for event ENDCRYPT	
PUBLISH_ERROR	0x188		Publish configuration for event ERROR	Deprecated
SHORTS	0x200		Shortcuts between local events and tasks	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
MICSTATUS	0x400		MIC check result	
ENABLE	0x500		Enable	
MODE	0x504		Operation mode	

Register	Offset	Security	Description
CNFPTR	0x508		Pointer to data structure holding the AES key and the NONCE vector
INPTR	0x50C		Input pointer
OUTPTR	0x510		Output pointer
SCRATCHPTR	0x514		Pointer to data area used for temporary storage
MAXPACKETSIZE	0x518		Length of keystream generated when MODE.LENGTH = Extended
RATEOVERRIDE	0x51C		Data rate override setting.
HEADERMASK	0x520		Header (S0) mask.

Table 60: Register overview

7.5.10.1 TASKS_KSGEN

Address offset: 0x000

Start generation of keystream. This operation will stop by itself when completed.

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value		Description
A	W	TASKS_KSGEN				Start generation of keystream. This operation will stop by itself when completed.
			Trigger	1		Trigger task

7.5.10.2 TASKS_CRYPT

Address offset: 0x004

Start encryption/decryption. This operation will stop by itself when completed.

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value		Description
A	W	TASKS_CRYPT				Start encryption/decryption. This operation will stop by itself when completed.
			Trigger	1		Trigger task

7.5.10.3 TASKS_STOP

Address offset: 0x008

Stop encryption/decryption

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value		Description
A	W	TASKS_STOP				Stop encryption/decryption
			Trigger	1		Trigger task

7.5.10.4 TASKS_RATEOVERRIDE

Address offset: 0x00C

Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RATEOVERRIDE			Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption
			Trigger	1	Trigger task

7.5.10.5 SUBSCRIBE_KSGEN

Address offset: 0x080

Subscribe configuration for task KSGEN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task KSGEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.5.10.6 SUBSCRIBE_CRYPT

Address offset: 0x084

Subscribe configuration for task CRYPT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CRYPT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.5.10.7 SUBSCRIBE_STOP

Address offset: 0x088

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.5.10.8 SUBSCRIBE_RATEOVERRIDE

Address offset: 0x08C

Subscribe configuration for task RATEOVERRIDE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RATEOVERRIDE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.5.10.9 EVENTS_ENDKSGEN

Address offset: 0x100

Keystream generation complete

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDKSGEN			Keystream generation complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.5.10.10 EVENTS_ENDCRYPT

Address offset: 0x104

Encrypt/decrypt complete

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDCRYPT			Encrypt/decrypt complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.5.10.11 EVENTS_ERROR (Deprecated)

Address offset: 0x108

CCM 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description	
A	RW	EVENTS_ERROR			CCM error event	Deprecated
			NotGenerated	0	Event not generated	
			Generated	1	Event generated	

7.5.10.12 PUBLISH_ENDKSGEN

Address offset: 0x180

Publish configuration for event ENDKSGEN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDKSGEN will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.5.10.13 PUBLISH_ENDCRYPT

Address offset: 0x184

Publish configuration for event ENDCRYPT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDCRYPT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.5.10.14 PUBLISH_ERROR (Deprecated)

Address offset: 0x188

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.5.10.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	ENDKSGEN_CRYPT			Shortcut between event ENDKSGEN and task CRYPT
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.5.10.16 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENDKSGEN			Write '1' to enable interrupt for event ENDKSGEN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDCRYPT			Write '1' to enable interrupt for event ENDCRYPT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ERROR			Write '1' to enable interrupt for event ERROR Deprecated
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.5.10.17 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENDKSGEN			Write '1' to disable interrupt for event ENDKSGEN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDCRYPT			Write '1' to disable interrupt for event ENDCRYPT
			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				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	R/W	Field	Value ID	Value	Description	
C	RW	ERROR			Write '1' to disable interrupt for event ERROR	Deprecated
			Clear	1	Disable	
			Disabled	0	Read: Disabled	
			Enabled	1	Read: Enabled	

7.5.10.18 MICSTATUS

Address offset: 0x400

MIC check result

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

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

7.5.10.19 ENABLE

Address offset: 0x500

Enable

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

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

7.5.10.20 MODE

Address offset: 0x504

Operation mode

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MODE			The mode of operation to be used. Settings in this register apply whenever either the KSGEN task or the CRYPT task is triggered.
			Encryption	0	AES CCM packet encryption mode
			Decryption	1	AES CCM packet decryption mode
B	RW	DATARATE			Radio data rate that the CCM shall run synchronous with
			1Mbit	0	1 Mbps

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
			2Mbit	1	2 Mbps
			125Kbps	2	125 kbps
			500Kbps	3	500 kbps
C	RW	LENGTH			Packet length configuration
			Default	0	Default length. Effective length of LENGTH field in encrypted/decrypted packet is 5 bits. A keystream for packet payloads up to 27 bytes will be generated.
			Extended	1	Extended length. Effective length of LENGTH field in encrypted/decrypted packet is 8 bits. A keystream for packet payloads up to MAXPACKETSIZE bytes will be generated.

7.5.10.21 CNFPTR

Address offset: 0x508

Pointer to data structure holding the AES key and the NONCE vector

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

7.5.10.22 INPTR

Address offset: 0x50C

Input pointer

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

7.5.10.23 OUTPTR

Address offset: 0x510

Output pointer

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

7.5.10.24 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SCRATCHPTR			Pointer to a scratch data area used for temporary storage during keystream generation, MIC generation and encryption/decryption. The scratch area is used for temporary storage of data during keystream generation and encryption. When MODE.LENGTH = Default, a space of 43 bytes is required for this temporary storage. When MODE.LENGTH = Extended, a space of (16 + MAXPACKETSIZE) bytes is required.

7.5.10.25 MAXPACKETSIZE

Address offset: 0x518

Length of keystream generated when MODE.LENGTH = Extended

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x000000FB				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXPACKETSIZE		[0x001B..0x00FB]	Length of keystream generated when MODE.LENGTH = Extended. This value must be greater than or equal to the subsequent packet payload to be encrypted/decrypted.

7.5.10.26 RATEOVERRIDE

Address offset: 0x51C

Data rate override setting.

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

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

7.5.10.27 HEADERMASK

Address offset: 0x520

Header (S0) mask.

Bitmask for packet header (S0) before MIC generation/authentication.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	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 0x000000E3	**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**	**1**

ID	R/W	Field	Value ID	Value	Description
A	RW	HEADERMASK			Header (S0) mask

7.5.11 Electrical specification

7.5.11.1 Timing specification

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

7.6 COMP — Comparator

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

The main features of COMP are the following:

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

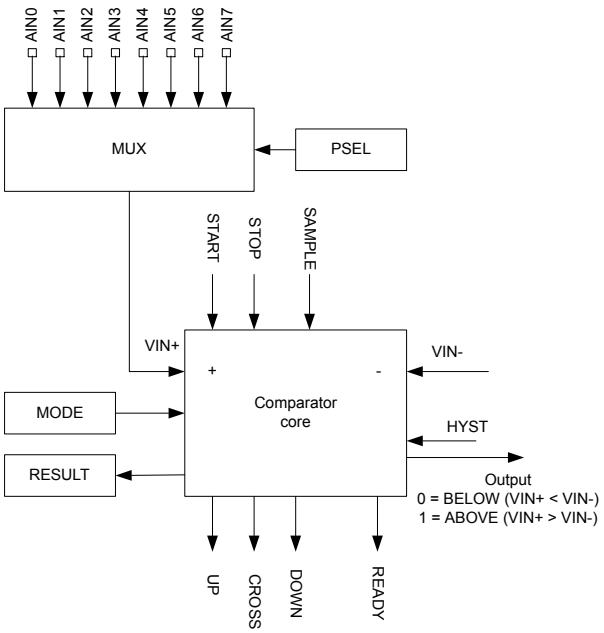

Figure 47: Comparator overview

Once enabled (using the ENABLE register), the comparator is started by triggering the START task and stopped by triggering the STOP task. The comparator will generate a READY event to indicate when it is ready for use and the output is correct. The delay between START and READY is $t_{INT\_REF,START}$ if an internal reference is selected, or $t_{COMP,START}$ if an external reference is used. When the COMP module is started, events will be generated every time VIN+ crosses VIN-.

Operation modes

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

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

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

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

An optional hysteresis on VIN+ and VIN- can be enabled when the module is used in differential mode through the HYST register. In single-ended mode, VUP and VDOWN thresholds can be set to implement a hysteresis using the reference ladder (see Comparator in single-ended mode on page 181). This hysteresis is in the order of magnitude of $V_{DIFFHYST}$, and shall prevent noise on the signal to create unwanted events. See Hysteresis example where VIN+ starts below VUP on page 181 for an illustration of the effect of an active hysteresis on a noisy input signal.

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

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

ISOURCE

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 µA, 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 using the following equation:

```
f_OSC = I_SOURCE / (2C·(VUP-VDOWN) )
```

7.6.1 Shared resources

The COMP shares analog resources with other analog peripherals.

While it is possible to use the SAADC at the same time as the COMP, selecting the same analog input pin for both peripherals is not supported.

Additionally, COMP shares registers and other resources with other peripherals that have the same ID as the COMP. See Peripherals with shared ID on page 147 for more information.

The COMP 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 behavior.

7.6.2 Differential mode

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

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

- PSEL
- MODE
- EXTREFSEL

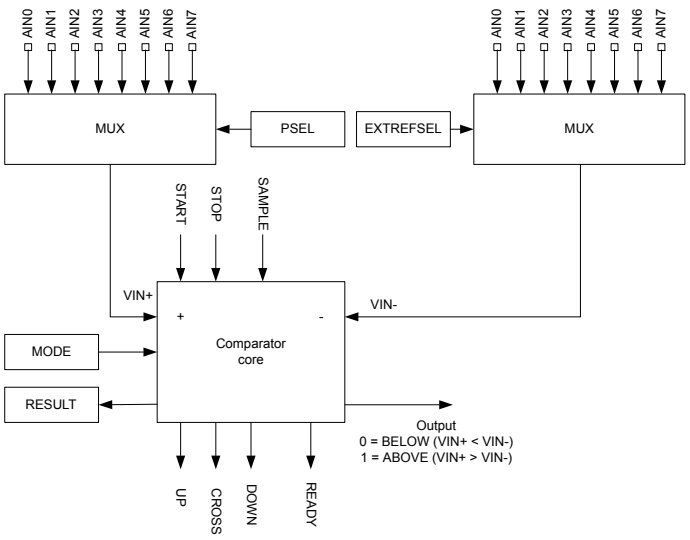

Figure 48: Comparator in differential mode

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

When the HYST register is turned on during this mode, the output of the comparator and associated events do the following:

- Change from ABOVE to BELOW when VIN+ drops below VIN- - ($V_{DIFFHYST}$/2)
- Change from BELOW to ABOVE when VIN+ raises above VIN- + ($V_{DIFFHYST}$/2)

This behavior is illustrated in the following figure.

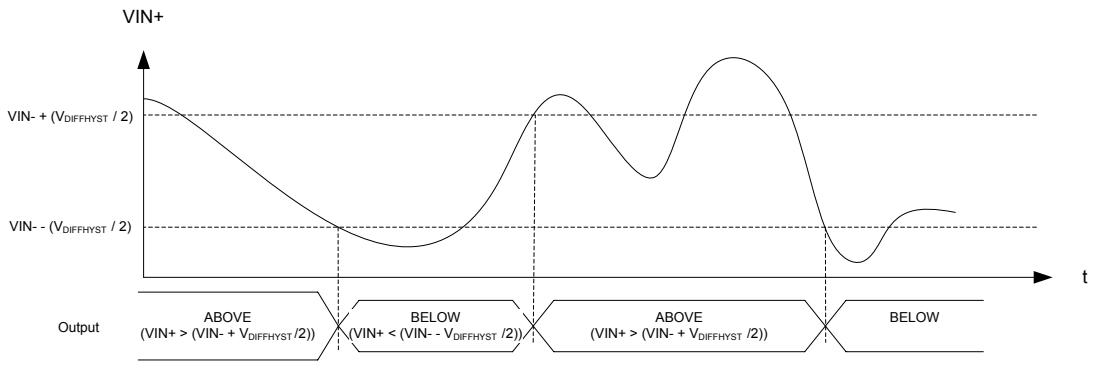

Figure 49: Hysteresis enabled in differential mode

7.6.3 Single-ended mode

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

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

- PSEL
- MODE
- REFSEL
- EXTREFSEL
- TH

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

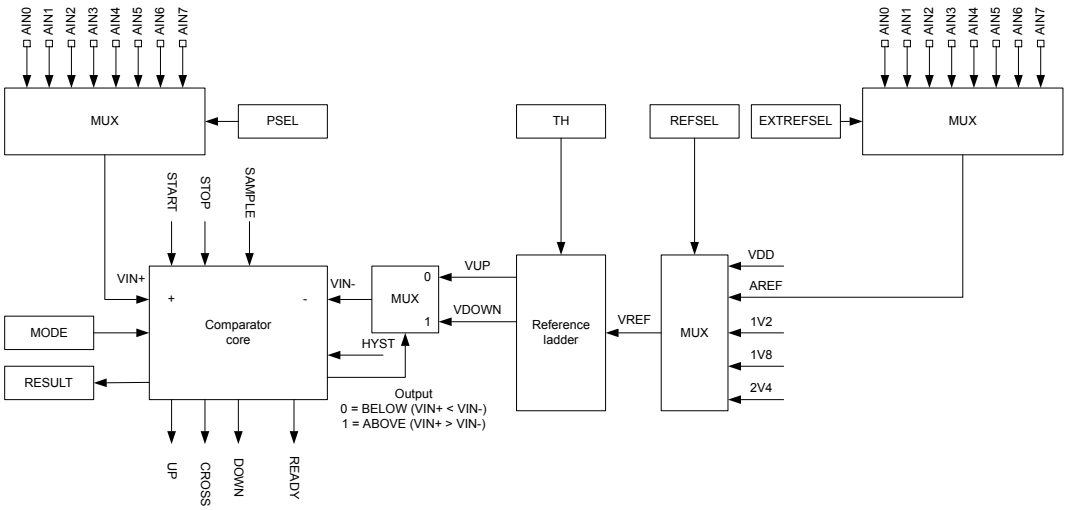

Figure 50: Comparator in single-ended mode

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

When the comparator core detects that VIN+ > VIN-, i.e. ABOVE as per the RESULT register, VIN- will switch to VDOWN. When VIN+ falls below VIN- again, VIN- will be switched back to VUP. By specifying VUP larger than VDOWN, a hysteresis can be generated as illustrated in the following figures.

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

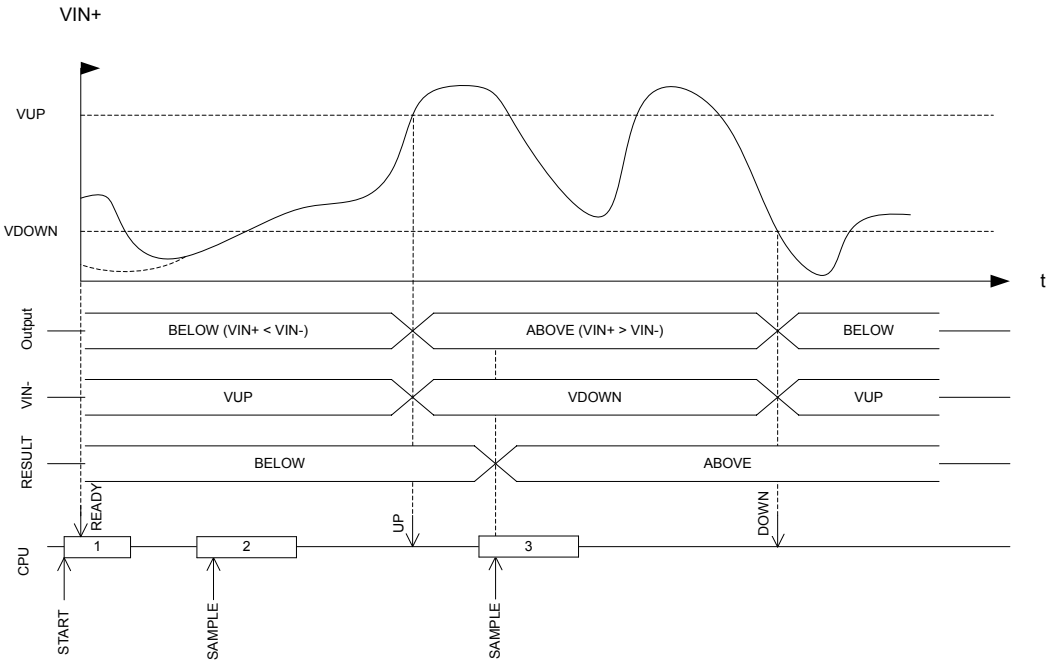

Figure 51: Hysteresis example where VIN+ starts below VUP

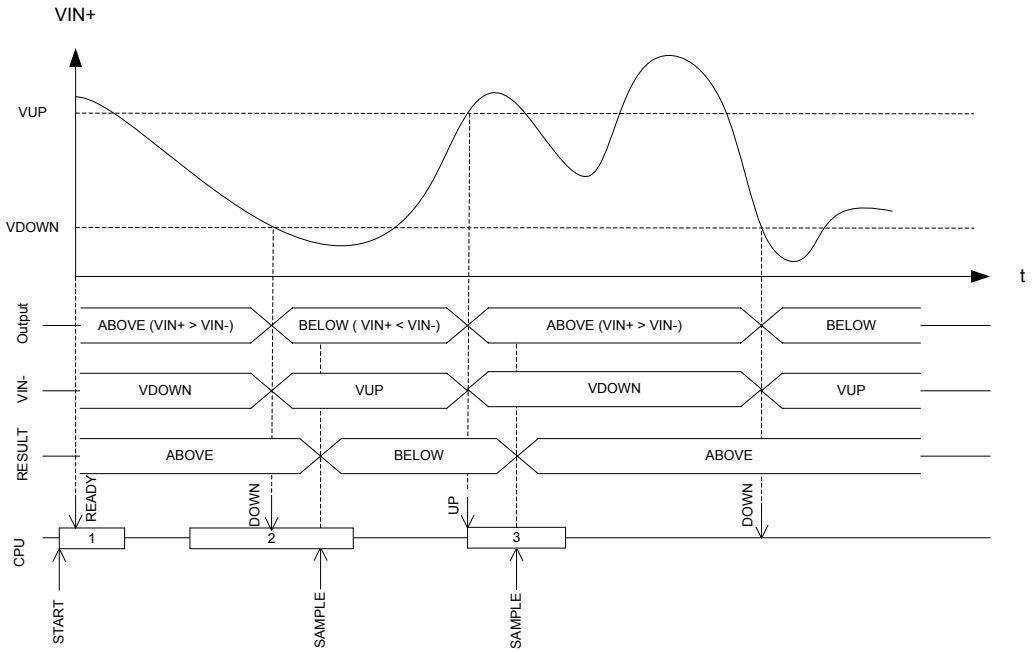

Figure 52: Hysteresis example where VIN+ starts above VUP

7.6.4 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5001A000	APPLICATION	COMP	COMP : S	US	NA	Comparator	
0x4001A000			COMP : NS				

Table 61: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start comparator
TASKS_STOP	0x004		Stop comparator
TASKS_SAMPLE	0x008		Sample comparator value
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
SUBSCRIBE_SAMPLE	0x088		Subscribe configuration for task SAMPLE
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
PUBLISH_READY	0x180		Publish configuration for event READY
PUBLISH_DOWN	0x184		Publish configuration for event DOWN
PUBLISH_UP	0x188		Publish configuration for event UP
PUBLISH_CROSS	0x18C		Publish configuration for event CROSS
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
RESULT	0x400		Compare result
ENABLE	0x500		COMP enable
PSEL	0x504		Pin select
REFSEL	0x508		Reference source select for single-ended mode

Register	Offset	Security	Description
EXTREFSEL	0x50C		External reference select
TH	0x530		Threshold configuration for hysteresis unit
MODE	0x534		Mode configuration
HYST	0x538		Comparator hysteresis enable
ISOURCE	0x53C		Current source select on analog input

Table 62: Register overview

7.6.4.1 TASKS_START

Address offset: 0x000

Start comparator

Bit number				31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value								Description																							
A	W	TASKS_START										Start comparator																							
			Trigger	1								Trigger task																							

7.6.4.2 TASKS_STOP

Address offset: 0x004

Stop comparator

Bit number				31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value								Description																							
A	W	TASKS_STOP										Stop comparator																							
			Trigger	1								Trigger task																							

7.6.4.3 TASKS_SAMPLE

Address offset: 0x008

Sample comparator 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value								Description																							
A	W	TASKS_SAMPLE										Sample comparator value																							
			Trigger	1								Trigger task																							

7.6.4.4 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.6.4.5 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.6.4.6 SUBSCRIBE_SAMPLE

Address offset: 0x088

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SAMPLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.6.4.7 EVENTS_READY

Address offset: 0x100

COMP is ready and output is valid

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_READY			COMP is ready and output is valid
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.6.4.8 EVENTS_DOWN

Address offset: 0x104

Downward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DOWN			Downward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.6.4.9 EVENTS_UP

Address offset: 0x108

Upward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_UP			Upward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.6.4.10 EVENTS_CROSS

Address offset: 0x10C

Downward or upward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CROSS			Downward or upward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.6.4.11 PUBLISH_READY

Address offset: 0x180

Publish configuration for event READY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.6.4.12 PUBLISH_DOWN

Address offset: 0x184

Publish configuration for event DOWN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DOWN will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.6.4.13 PUBLISH_UP

Address offset: 0x188

Publish configuration for event UP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event UP will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.6.4.14 PUBLISH_CROSS

Address offset: 0x18C

Publish configuration for event CROSS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CROSS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.6.4.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

7.6.4.16 INTEN

Address offset: 0x300

Enable or disable interrupt

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

7.6.4.17 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for event READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to enable interrupt for event DOWN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to enable interrupt for event UP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to enable interrupt for event CROSS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.6.4.18 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to disable interrupt for event DOWN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to disable interrupt for event UP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to disable interrupt for event CROSS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.6.4.19 RESULT

Address offset: 0x400

Compare result

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

7.6.4.20 ENABLE

Address offset: 0x500

COMP enable

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

7.6.4.21 PSEL

Address offset: 0x504

Pin select

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

7.6.4.22 REFSEL

Address offset: 0x508

Reference source select for single-ended mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	REFSEL			Reference select
			Int1V2	0	VREF = internal 1.2 V reference (VDD >= 1.7 V)
			Int1V8	1	VREF = internal 1.8 V reference (VDD >= VREF + 0.2 V)
			Int2V4	2	VREF = internal 2.4 V reference (VDD >= VREF + 0.2 V)
			VDD	4	VREF = VDD
			ARef	5	VREF = AREF

7.6.4.23 EXTREFSEL

Address offset: 0x50C

External reference select

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EXTREFSEL			External analog reference select
			AnalogReference0	0	Use AIN0 as external analog reference
			AnalogReference1	1	Use AIN1 as external analog reference
			AnalogReference2	2	Use AIN2 as external analog reference
			AnalogReference3	3	Use AIN3 as external analog reference
			AnalogReference4	4	Use AIN4 as external analog reference
			AnalogReference5	5	Use AIN5 as external analog reference
			AnalogReference6	6	Use AIN6 as external analog reference
			AnalogReference7	7	Use AIN7 as external analog reference

7.6.4.24 TH

Address offset: 0x530

Threshold configuration for hysteresis unit

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

ID	R/W	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

7.6.4.25 MODE

Address offset: 0x534

Mode configuration

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

7.6.4.26 HYST

Address offset: 0x538

Comparator hysteresis enable

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

7.6.4.27 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				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ISOURCE			Comparator hysteresis
			Off	0	Current source disabled
			Ien2mA5	1	Current source enabled (+/- 2.5 µA)
			Ien5mA	2	Current source enabled (+/- 5 µA)
			Ien10mA	3	Current source enabled (+/- 10 µA)

7.6.5 Electrical specification

7.6.5.1 COMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{PROPDLY,LP}$	Propagation delay, Low-power mode[9]		0.6		µs
$t_{PROPDLY,N}$	Propagation delay, Normal mode[9]		0.2		µs
$t_{PROPDLY,HS}$	Propagation delay, High-speed mode[9]		0.1		µs

[9] Propagation delay is with 10 mV overdrive.

Symbol	Description	Min.	Typ.	Max.	Units
I_{SOURCE}	Configurable input current provided by the output driven current source.	..	..	..	µA
$I_{SOURCE,A}$		..	..	..	µA
$I_{SOURCE,B}$		..	..	..	µA
$I_{SOURCE,C}$		..	..	..	µA
$V_{DIFFHYST}$	Optional hysteresis applied to differential input	26	35	55	mV
$V_{VDD-VREF}$	Required difference between VDD and a selected VREF, VDD > VREF	0.3			V
$t_{INT\_REF,START}$	Startup time for the internal bandgap reference		50	80	µs
E_{INT_REF}	Internal bandgap reference error	-1.8	0	0.5	%
$V_{INPUTOFFSET}$	Input offset	-10		10	mV
$t_{COMP,START}$	Startup time for the comparator core		3		µs

7.7 CRYPTOCELL — Arm TrustZone CryptoCell 312

Arm TrustZone CryptoCell 312 (CRYPTOCELL) is a security subsystem providing root of trust (RoT) and cryptographic services for a device.

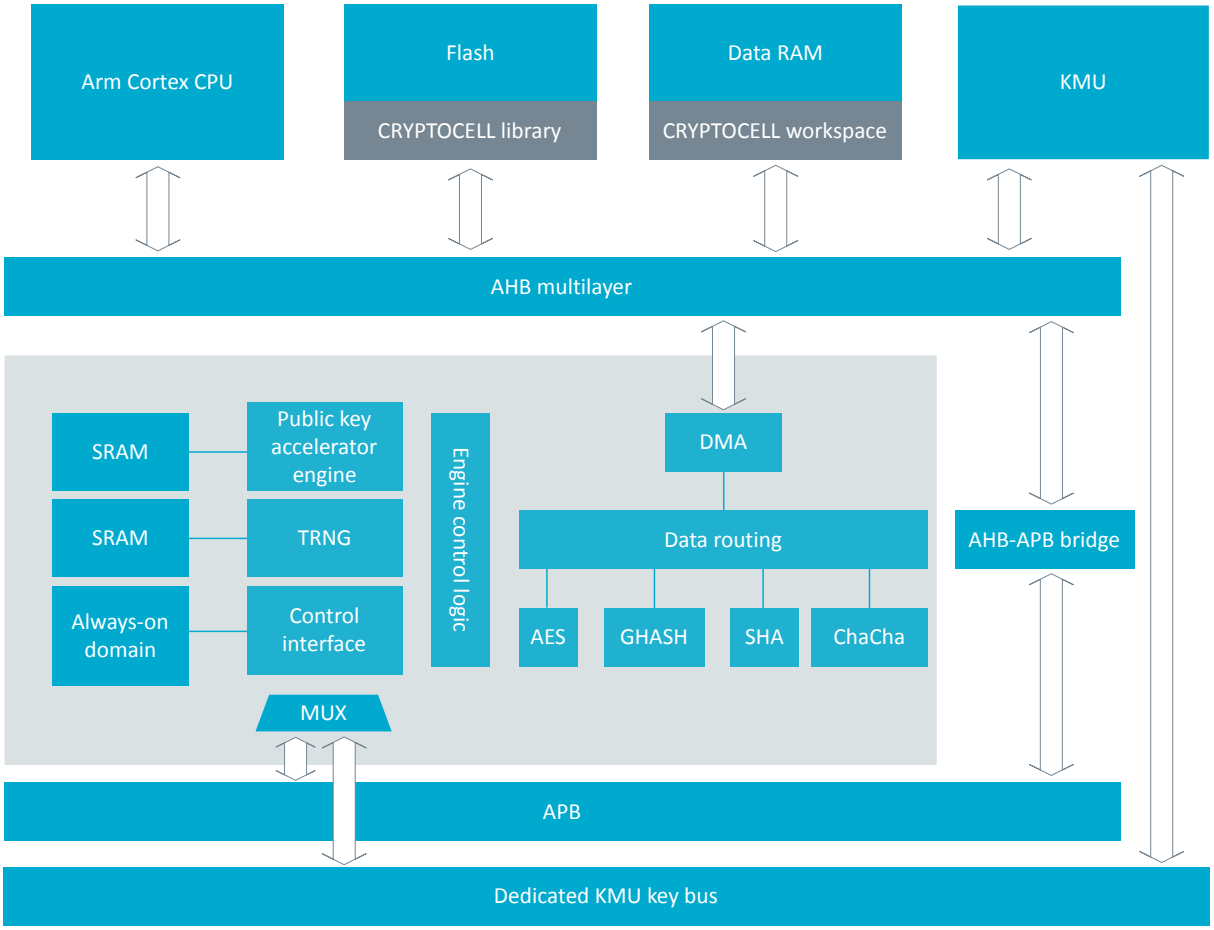

Figure 53: CRYPTOCELL block diagram

The following cryptographic features are provided:

- True random number generator (TRNG) compliant with NIST 800-90B, AIS-31, and FIPS 140-2
- Pseudorandom number generator (PRNG) using underlying AES engine compliant with NIST 800-90A
- RSA public key cryptography
  - Up to 3072-bit key size

  - PKCS#1 v2.1/v1.5
  - Optional CRT support
- Elliptic curve cryptography (ECC)
  - NIST FIPS 186-4 recommended prime field curves using pseudorandom parameters, up to 521 bits:
    - P-192
    - P-224
    - P-256
    - P-384
    - P-521
  - SEC 2 recommended prime field curves using pseudorandom parameters, up to 521 bits:
    - secp160r1
    - secp192r1
    - secp224r1
    - secp256r1
    - secp384r1
    - secp521r1
  - Koblitz prime field curves using fixed parameters, up to 256 bits:
    - secp160k1
    - secp192k1
    - secp224k1
    - secp256k1
  - Edwards/Montgomery curves:
    - Ed25519
    - Curve25519
  - ECDH/ECDSA support
- Secure remote password protocol (SRP) with up to 3072-bit operations
- Hashing functions
  - SHA-1
  - SHA-2 up to 256 bits
  - Keyed-hash message authentication code (HMAC)
- AES symmetric encryption
  - General purpose AES engine (encrypt/decrypt, sign/verify)
  - Supported key size - 128 and 256 bits
  - Supported encryption modes: ECB, CBC, CMAC/CBC-MAC, CTR, CCM/CCM*, GCM
- ChaCha20/Poly1305 symmetric encryption
  - Supported key size - 128 and 256 bits
  - Authenticated encryption with associated data (AEAD) mode

7.7.1 Usage

The CRYPTOCELL state is controlled via a register interface. The CRYPTOCELL cryptographic functions are accessible through a software library provided in the device SDK.

To enable CRYPTOCELL, use register ENABLE on page 196.

> **Note:** To prevent the device from reaching the System ON All Idle state, enable the CRYPTOCELL subsystem.

7.7.2 Direct memory access (DMA)

CRYPTOCELL features direct access memory (DMA) to allow cryptographic operations on memory mapped regions without involving the CPU.

The maximum DMA transaction size is limited to 2^{16}-1 bytes. See Memory on page 18 for information about memory that is accessible through the CRYPTOCELL DMA.

The CRYPTOCELL DMA can configure the security setting used for bus transactions.

Any data stored in a memory type not accessible by the CRYPTOCELL DMA engine must be copied to a memory type accessible by the direct memory before it can be processed by the CRYPTOCELL subsystem.

7.7.3 Standards

Arm TrustZone CryptoCell 312 (CRYPTOCELL) is compliant with the protocol specifications and standards shown in the following table.

Algorithm family	Identification code	Document title
TRNG	NIST SP 800-90B	*Recommendation for the Entropy Sources Used for Random Bit Generation*
	AIS-31	*A proposal for: Functionality classes and evaluation methodology for physical random number generators*
	FIPS 140-2	*Security Requirements for Cryptographic Modules*
PRNG	NIST SP 800-90A	*Recommendation for Random Number Generation Using Deterministic Random Bit Generators*
Stream cipher	Chacha	*ChaCha, a variant of Salsa20*, Daniel J. Bernstein, January 28th 2008
MAC	Poly1305	*The Poly1305-AES message-authentication code*, Daniel J. Bernstein *Cryptography in NaCl*, Daniel J. Bernstein
Key agreement	SRP	*The Secure Remote Password Protocol*, Thomas Wu, November 11th 1997
Key derivation	NIST SP 800-108	*Recommendation for Key Derivation Using Pseudorandom Functions*. Compliant with section 5.1
AES	FIPS-197	*Advanced Encryption Standard (AES)*. Compliant with 128-bit and 256-bit key size only
	NIST SP 800-38A	*Recommendation for Block Cipher Modes of Operation - Methods and Techniques*. Compliant with sections 6.1, 6.2, 6.4, and 6.5.
	NIST SP 800-38B	*Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*
	NIST SP 800-38C	*Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*
	ISO/IEC 9797-1	AES CBC-MAC per ISO/IEC 9797-1 MAC algorithm 1
	IEEE 802.15.4-2011	*IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Annex B.4: *Specification of generic CCM* mode of operation*
Hash	FIPS 180-3	Secure Hash Standard (SHA1, SHA-224, SHA-256)
	RFC2104	*HMAC: Keyed-Hashing for Message Authentication*
RSA	PKCS#1	*Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications* v1.5/2.1
Diffie-Hellman	ANSI X9.42	*Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*
	PKCS#3	*Diffie-Hellman Key-Agreement Standard*
ECC	ANSI X9.63	*Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography*
	IEEE 1363	*Standard Specifications for Public-Key Cryptography*
	ANSI X9.62	*Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*
	Ed25519	Edwards-curve, *Ed25519: high-speed high-security signatures*, Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang
	Curve25519	Montgomery curve, *Curve25519: new Diffie-Hellman speed records*, Daniel J. Bernstein
	FIPS 186-4	*Digital Signature Standard (DSS)*. Compliant with sections 5.1, 6.2, 6.3, 6.4, B.1.2, B.2.2, B.3.6, B.4.2, C.3.1, C.3.3, C.3.5, C.9, and D.1.2.
	SEC 2	*Recommended Elliptic Curve Domain Parameters*, Certicom Research
	NIST SP 800-56A rev. 2	*Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*

Table 63: CRYPTOCELL cryptography standards

7.7.4 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50844000	APPLICATION	CRYPTOCELL	CRYPTOCELL	S	NSA	CryptoCell subsystem control interface	

Table 64: Instances

Register	Offset	Security	Description
ENABLE	0x500		Enable CRYPTOCELL subsystem.

Table 65: Register overview

7.7.4.1 ENABLE

Address offset: 0x500

Enable CRYPTOCELL subsystem.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable the CRYPTOCELL subsystem.
			Disabled	0	CRYPTOCELL subsystem disabled.
			Enabled	1	CRYPTOCELL subsystem enabled. When enabled, the CRYPTOCELL subsystem can be initialized and controlled through the CryptoCell firmware API.

7.8 DCNF — Domain configuration

The domain configuration (DCNF) module provides a way to identify the CPU by its CPU ID in the device (CPUID). It also provides protection of the AHB multilayer interconnect (AMLI).

To provide for the AMLI protection, the DCNF contains configuration registers that can be used to block some paths from the AHB masters to their respective AHB slaves in the AMLI.

For an illustration of how the AHB masters and slaves are connected through the AMLI, see Memory on page 18.

7.8.1 Protection

The DCNF protection involves blocking of paths from AHB masters in an external core to the AHB slaves in the local core's AMLI. This way, the local core's internal resources can be blocked from being accessed by an external core. A set of configuration registers is used to control this behavior.

See Memory on page 18 to get an overview of the AMLI.

The DCNF configuration registers that enable the DCNF protection are the following:

- EXTPERI[n].PROTECT
- EXTRAM[n].PROTECT
- EXTCODE[n].PROTECT

An attempt to access the blocked resources will trigger a BusFault or a HardFault exception, depending on the value of the BUSFAULTENA bit in the Arm Cortex-M33 SHCSR (system handler control and state register), described in the Arm Cortex-M33 Devices Generic User Guide.

RAM protection

The protection of RAM regions is configured through the SLAVE-bits of the corresponding master ports' register EXTRAM[n].PROTECT.

Peripheral protection

The protection of peripheral memory regions is configured through the SLAVE-bits of the corresponding master ports' register EXTPERI[n].PROTECT.

Code protection

The protection of code memory regions is configured through the SLAVE-bits of the corresponding master ports' register EXTCODE[0].PROTECT.

7.8.2 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50000000 0x40000000	APPLICATION	DCNF	DCNF : S DCNF : NS	US	NA	Domain configuration	CPUID value is 0x00000000
0x41000000	NETWORK	DCNF	DCNF	NS	NA	Domain configuration	Registers EXTPERI[n].PROTECT, EXTRAM[n].PROTECT, and EXTCODE[n].PROTECT not available for the network core. CPUID value is 0x00000001

Table 66: Instances

Register	Offset	Security	Description
CPUID	0x420		CPU ID of this subsystem
EXTPERI[n].PROTECT	0x440		Control access for master connected to AMLI master port EXTPERI[n]
EXTRAM[n].PROTECT	0x460		Control access from master connected to AMLI master port EXTRAM[n]
EXTCODE[n].PROTECT	0x480		Control access from master connected to AMLI master port EXTCODE[n]

Table 67: Register overview

7.8.2.1 CPUID

Address offset: 0x420

CPU ID of this subsystem

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	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	R/W	Field	Value ID	Value	Description
A	R	CPUID			CPU ID

7.8.2.2 EXTPERI[n].PROTECT (n=0..0)

Address offset: 0x440 + (n × 0x4)

Control access for master connected to AMLI master port EXTPERI[n]

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A-A	RW	SLAVE[i] (i=0..0)			Control access to slave i of master EXTPERI[n]
			Allowed	0	Access to slave is allowed
			Blocked	1	Access to slave is blocked

7.8.2.3 EXTRAM[n].PROTECT (n=0..0)

Address offset: 0x460 + (n × 0x4)

Control access from master connected to AMLI master port EXTRAM[n]

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

ID	R/W	Field	Value ID	Value	Description
A-H	RW	SLAVE[i] (i=0..7)			Control access to slave i of master EXTRAM[n]
			Allowed	0	Access to slave is allowed
			Blocked	1	Access to slave is blocked

7.8.2.4 EXTCODE[n].PROTECT (n=0..0)

Address offset: 0x480 + (n × 0x4)

Control access from master connected to AMLI master port EXTCODE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A-A	RW	SLAVE[i] (i=0..0)			Control access to slave i of master EXTCODE[n]
			Allowed	0	Access to slave is allowed
			Blocked	1	Access to slave is blocked

7.9 DPPI - Distributed programmable peripheral interconnect

The distributed programmable peripheral interconnect (DPPI) enables peripherals to interact autonomously with each other by using tasks and events, without any intervention from the CPU. DPPI allows precise synchronization between peripherals when real-time application constraints exist, and eliminates the need for CPU involvement to implement behavior which can be predefined using the DPPI.

Note: For more information on tasks, events, publish/subscribe, interrupts, and other concepts, see Peripheral interface on page 145.

The DPPI has the following features:

- Peripheral tasks can subscribe to channels
- Peripheral events can be published on channels
- Publish/subscribe pattern enabling multiple connection options that include the following:
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many

The DPPI consists of several PPIBus modules, which are connected to a fixed number of DPPI channels and a DPPI configuration (DPPIC).

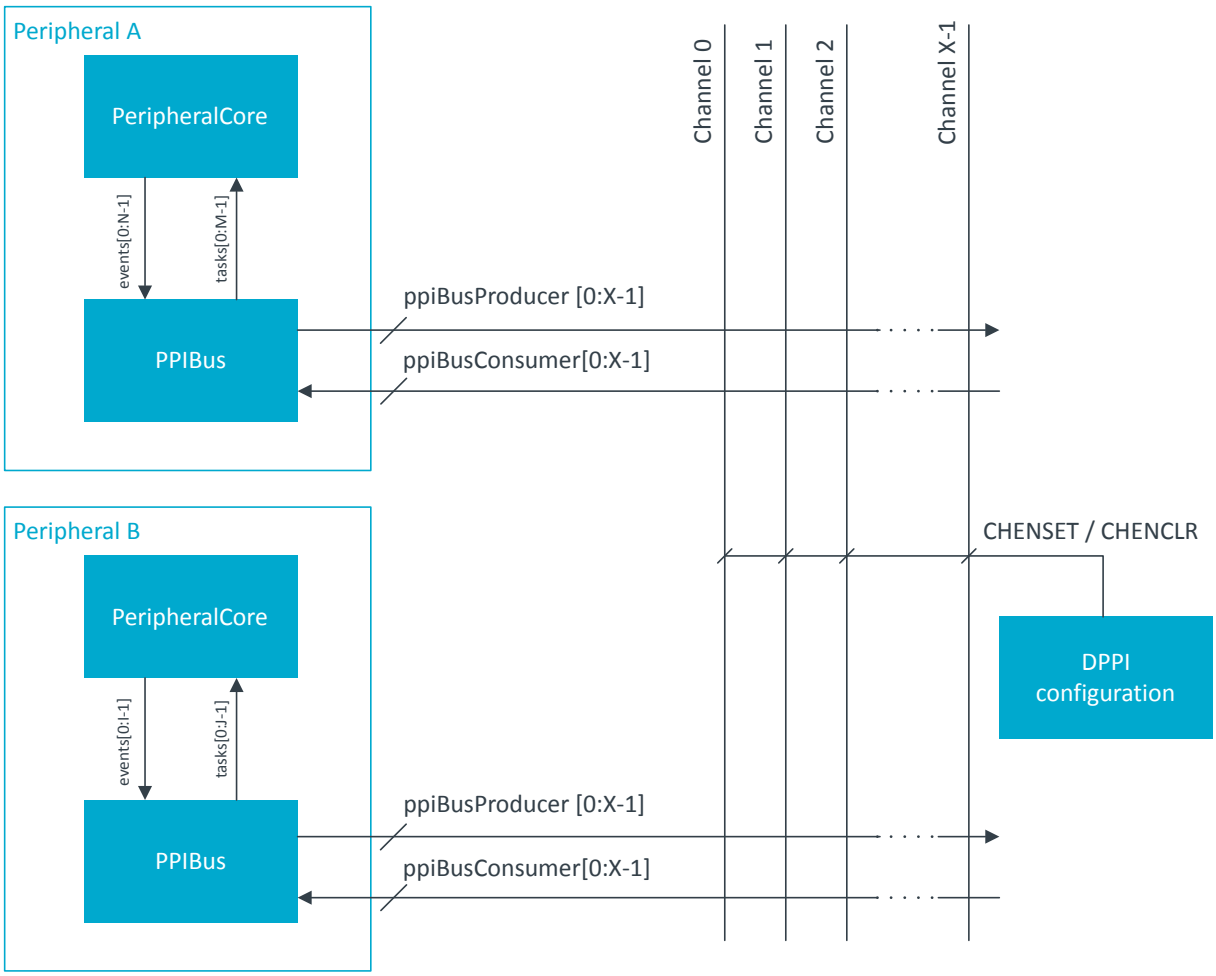

Figure 54: DPPI overview

7.9.1 Subscribing to and publishing on channels

The PPIBus can route peripheral events onto the channels (publishing), or route events from the channels into peripheral tasks (subscribing).

All peripherals include the following:

- One subscribe register per task
- One publish register per event

Publish and subscribe registers use a channel index field to determine the channel to which the event is published or tasks subscribed. In addition, there is an enable bit for the subscribe and publish registers that needs to be enabled before the subscription or publishing takes effect.

Writing non-existing channel index (CHIDX) numbers into a peripheral's publish or subscribe registers will yield unexpected results.

One event can trigger multiple tasks by subscribing different tasks to the same channel. Similarly, one task can be triggered by multiple events by publishing different events to the same channel. For advanced use cases, multiple events and multiple tasks can connect to the same channel forming a many-to-many connection. If multiple events are published on the same channel at the same time, the events are merged and only one event is routed through the DPPI.

How peripheral events are routed onto different channels based on publish registers is illustrated in the following figure.

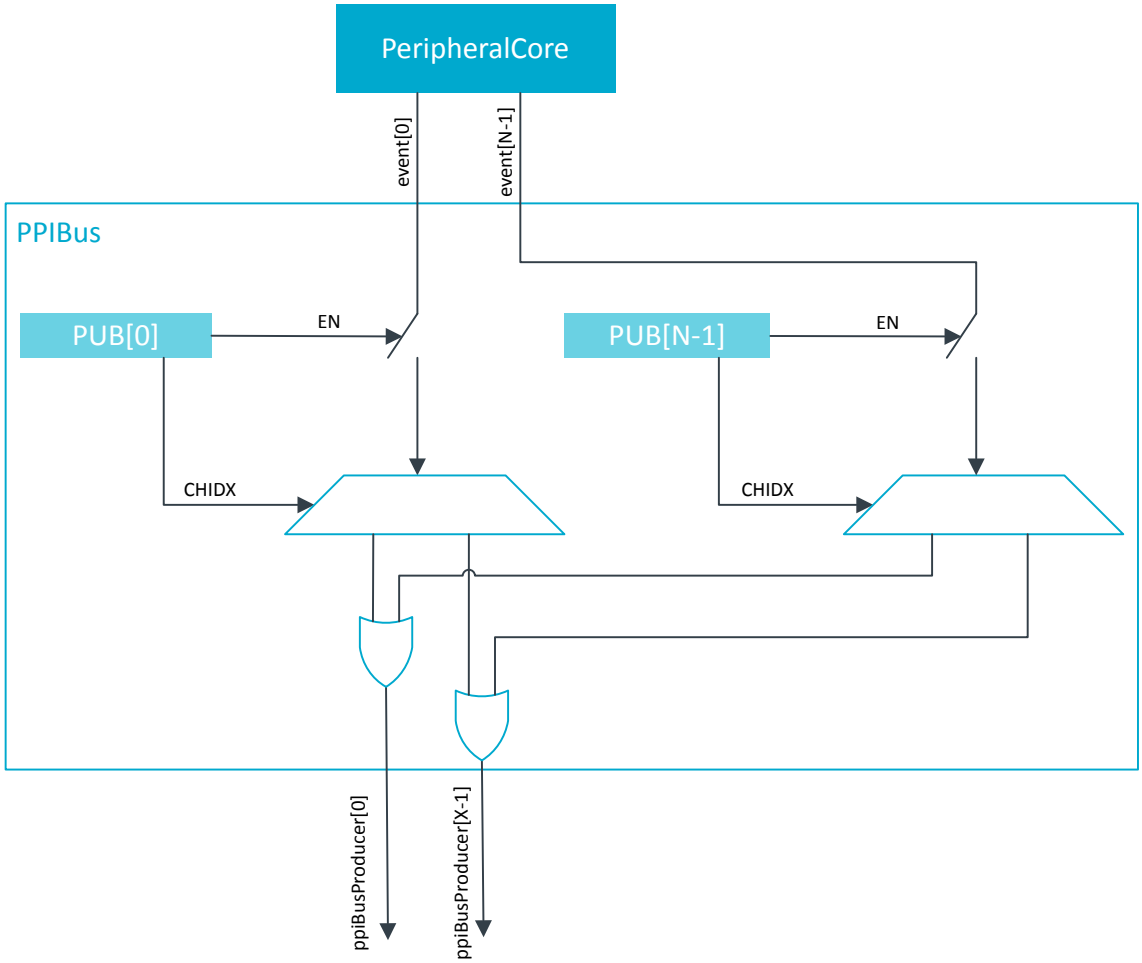

Figure 55: DPPI events flow

The following figure illustrates how peripheral tasks are triggered from different channels based on subscribe registers.

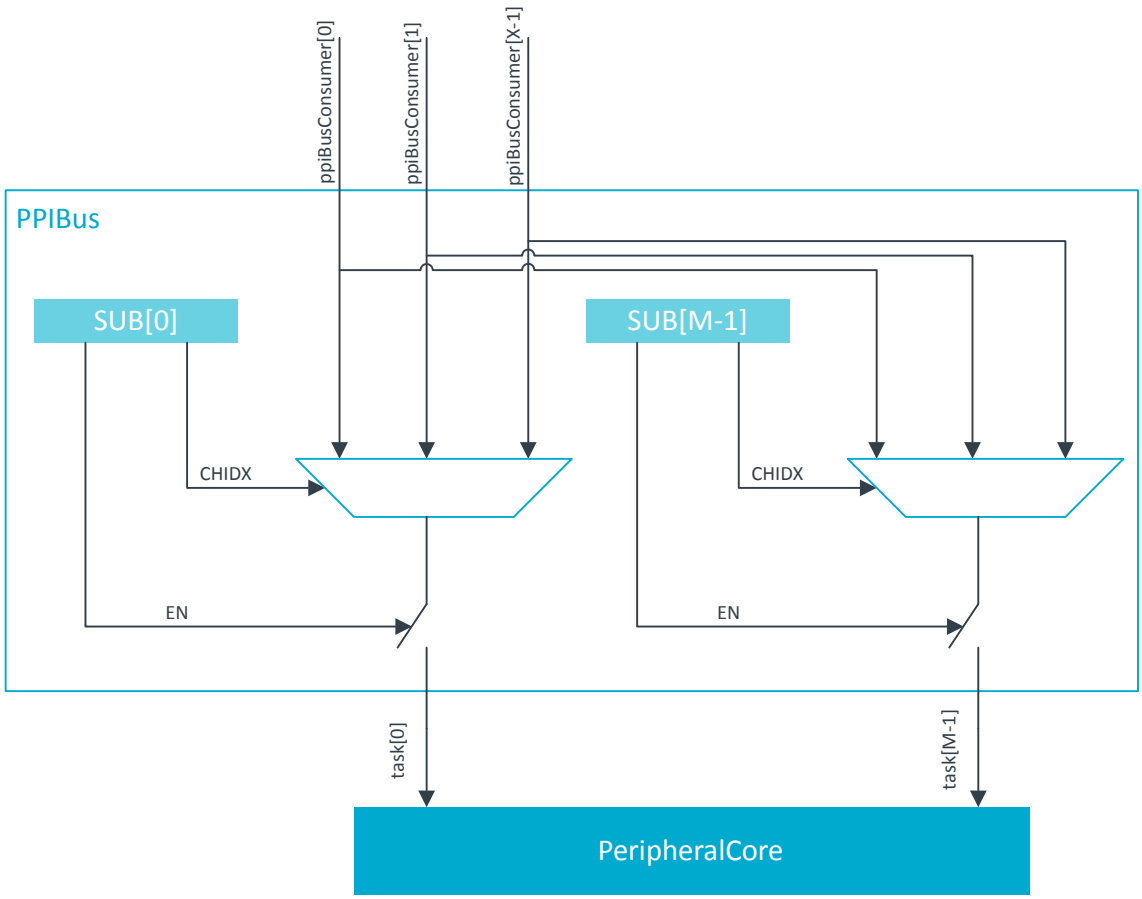

Figure 56: DPPI tasks flow

7.9.2 DPPI configuration (DPPIC)

Enabling and disabling of channels globally is handled through the DPPI configuration (DPPIC). Connection (connect/disconnect) between a channel and a peripheral is handled locally by the PPIBus.

There are two ways of enabling and disabling global channels using the DPPI configuration:

- Enable or disable channels individually using registers CHEN, CHENSET, and CHENCLR.
- Enable or disable channels in channel groups using the groups' tasks ENABLE and DISABLE. It needs to be defined which channels belong to which channel groups before these tasks are triggered.

Note: ENABLE tasks are prioritized over DISABLE tasks. When a channel belongs to two or more groups, for example group m and n, and the tasks CHG[m].EN and CHG[n].DIS occur simultaneously (m and n can be equal or different), the CHG[m].EN task on that channel is prioritized.

The DPPI configuration tasks (for example CHG[0].EN) can be triggered through DPPI like any other task, which means they can be linked to a DPPI channel through the subscribe registers.

In order to write to CHG[x], the corresponding CHG[x].EN and CHG[x].DIS subscribe registers must be disabled. Writes to CHG[x] are ignored if any of the two subscribe registers are enabled.

7.9.3 Connection examples

DPPI offers several connection options. Examples are given for how to create one-to-one and many-to-many connections.

One-to-one connection

This example shows how to create a one-to-one connection between TIMER compare register and SAADC start task.

The channel configuration is set up first. TIMER0 will publish its COMPARE0 event on channel 0, and SAADC will subscribe its START task to events on the same channel. After that, the channel is enabled through the DPPIC.

```
NRF_TIMER0->PUBLISH_COMPARE0 = (DPPI_PUB_CHIDX_Ch0) |
                               (DPPI_PUB_EN_Msk);
NRF_SAADC->SUBSCRIBE_START   = (DPPI_SUB_CHIDX_Ch0) |
                               (DPPI_SUB_EN_Msk);

NRF_DPPIC->CHENSET = (DPPI_CHENSET_CH0_Set << DPPI_CHENSET_CH0_Pos);
```

Many-to-many connection

The example shows how to create a many-to-many connection, showcasing the DPPIC's channel group functionality.

A channel group that includes only channel 0 is set up first. Then the GPIOTE and TIMER0 configure their IN0 and COMPARE0 events respectively to be published on channel 0, while the SAADC configures its START task to subscribe to events on channel 0. Through DPPIC, the CHG0 DISABLE task is configured to subscribe to events on channel 0. After an event is received on channel 0 it will be disabled. Finally, channel 0 is enabled using the DPPIC task to enable a channel group.

```
NRF_DPPIC->CHG[0] = (DPPI_CHG_CH0_Included << PPI_CHG_CH0_Pos);

NRF_GPIOTE->PUBLISH_IN0         = (DPPI_PUB_CHIDX_Ch0) |
                                  (DPPI_PUB_EN_Msk);
NRF_TIMER0->PUBLISH_COMPARE0    = (DPPI_PUB_CHIDX_Ch0) |
                                  (DPPI_PUB_EN_Msk);
NRF_SAADC->SUBSCRIBE_START      = (DPPI_SUB_CHIDX_Ch0) |
                                  (DPPI_SUB_EN_Msk);
NRF_DPPIC->SUBSCRIBE_CHG[0].DIS = (DPPI_SUB_CHIDX_Ch0) |
                                  (DPPI_SUB_EN_Msk);

NRF_DPPIC->TASK_CHG[0].EN = 1;
```

7.9.4 Special considerations for a system implementing TrustZone for Cortex-M processors

DPPI is implemented with split security, meaning it handles both secure and non-secure accesses. In a system implementing the TrustZone for Cortex-M technology, DPPI channels can be defined as secure or non-secure using the SPU.

A peripheral configured as non-secure will only be able to subscribe to or publish on non-secure DPPI channels. A peripheral configured as secure will be able to access all DPPI channels. DPPI handles both secure and non-secure accesses, but behaves differently depending on the access type:

- A non-secure peripheral access can only configure and control the DPPI channels defined as non-secure in the SPU.DPPI.PERM[] register(s)

- A secure peripheral access can control all the DPPI channels, independently of the SPU.DPPI.PERM[] register(s)

A group of channels can be created, making it possible to simultaneously enable or disable all channels within the group. The security attribute of a group of channels (secure or non-secure) is defined as follows:

- If all channels (enabled or not) within a group are non-secure, then the group is considered non-secure
- If at least one of the channels (enabled or not) within the group is secure, then the group is considered secure

A non-secure access to a DPPI register, or a bit field, controlling a channel marked as secure in SPU.DPPI[].PERM register(s) will be ignored. Write accesses will have no effect, and read accesses will always return a zero value.

No exceptions are triggered when non-secure accesses target a register or a bit field controlling a secure channel. For example, if the bit `i` is set in the SPU.DPPI[0].PERM register (declaring DPPI channel i as secure), then:

- Non-secure write accesses to registers CHEN, CHENSET, and CHENCLR cannot write bit `i` of these registers
- Non-secure write accesses to TASK_CHG[j].EN and TASK_CHG[j].DIS registers are ignored if the channel group `j` contains at least one channel defined as secure (it can be the channel i itself or any channel declared as secure)
- Non-secure read accesses to registers CHEN, CHENSET, and CHENCLR always read 0 for the bit at position `i`

For the channel configuration registers (CHG[]), access from non-secure code is only possible if the included channels are all non-secure, whether the channels are enabled or not. If a CHG[g] register included one or more secure channel(s), then the group `g` is considered as secure, and only secure transfers can read to or write from CHG[g]. A non-secure write access is ignored, and a non-secure read access returns 0.

The DPPI can subscribe to secure and non-secure channels through the SUBSCRIBE_CHG[] registers, in order to trigger the task for enabling or disabling groups of channels. An event from a secure channel will be ignored if the group subscribing to this channel is non-secure. A secure group can subscribe to a non-secure channel or a secure channel.

7.9.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50017000 0x40017000	APPLICATION	DPPIC	DPPIC : S DPPIC : NS	SPLIT	NA	DPPI controller	Application core implements 32 DPPI channels
0x4100F000	NETWORK	DPPIC	DPPIC	NS	NA	DPPI controller	Network core implements 16 DPPI channels

Table 68: Instances

Register	Offset	Security	Description
TASKS_CHG[n].EN	0x000		Enable channel group n
TASKS_CHG[n].DIS	0x004		Disable channel group n
SUBSCRIBE_CHG[n].EN	0x080		Subscribe configuration for task CHG[n].EN
SUBSCRIBE_CHG[n].DIS	0x084		Subscribe configuration for task CHG[n].DIS
CHEN	0x500		Channel enable register
CHENSET	0x504		Channel enable set register
CHENCLR	0x508		Channel enable clear register
CHG[n]	0x800		Channel group n Note: Writes to this register are ignored if either SUBSCRIBE_CHG[n].EN or SUBSCRIBE_CHG[n].DIS is enabled

Table 69: Register overview

7.9.5.1 TASKS_CHG[n].EN (n=0..5)

Address offset: 0x000 + (n × 0x8)

Enable channel group n

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	W	EN			Enable channel group n
			Trigger	1	Trigger task

7.9.5.2 TASKS_CHG[n].DIS (n=0..5)

Address offset: 0x004 + (n × 0x8)

Disable channel group n

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	W	DIS			Disable channel group n
			Trigger	1	Trigger task

7.9.5.3 SUBSCRIBE_CHG[n].EN (n=0..5)

Address offset: 0x080 + (n × 0x8)

Subscribe configuration for task CHG[n].EN

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID									B																A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CHG[n].EN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.9.5.4 SUBSCRIBE_CHG[n].DIS (n=0..5)

Address offset: 0x084 + (n × 0x8)

Subscribe configuration for task CHG[n].DIS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CHG[n].DIS will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.9.5.5 CHEN

Address offset: 0x500

Channel enable register

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

7.9.5.6 CHENSET

Address offset: 0x504

Channel enable set register

Note: Read: Reads value of CH[i] field in CHEN register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Channel i enable set register. Writing 0 has no effect.
			Disabled	0	Read: Channel disabled
			Enabled	1	Read: Channel enabled
			Set	1	Write: Enable channel

7.9.5.7 CHENCLR

Address offset: 0x508

Channel enable clear register

Note: Read: Reads value of CH[i] field in CHEN register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-f	RW	CH[i] (i=0..31)			Channel i enable clear register. Writing 0 has no effect.
			Disabled	0	Read: Channel disabled
			Enabled	1	Read: Channel enabled
			Clear	1	Write: Disable channel

7.9.5.8 CHG[n] (n=0..5)

Address offset: 0x800 + (n × 0x4)

Channel group n

Note: Writes to this register are ignored if either SUBSCRIBE_CHG[n].EN or SUBSCRIBE_CHG[n].DIS is enabled

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

7.10 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.

7.10.1 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.

7.10.2 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 18 for more information about the different memory regions.

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

7.10.3 ECB data structure

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

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

Table 70: ECB data structure overview

7.10.4 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x4100D000	NETWORK	ECB	ECB	NS	NA	AES electronic code book (ECB) mode block encryption	

Table 71: Instances

Register	Offset	Security	Description
TASKS_STARTECB	0x000		Start ECB block encrypt
TASKS_STOPECB	0x004		Abort a possible executing ECB operation
SUBSCRIBE_STARTECB	0x080		Subscribe configuration for task STARTECB
SUBSCRIBE_STOPECB	0x084		Subscribe configuration for task STOPECB
EVENTS_ENDECB	0x100		ECB block encrypt complete
EVENTS_ERRORECB	0x104		ECB block encrypt aborted because of a STOPECB task or due to an error
PUBLISH_ENDECB	0x180		Publish configuration for event ENDECB
PUBLISH_ERRORECB	0x184		Publish configuration for event ERRORECB
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ECBDATAPTR	0x504		ECB block encrypt memory pointers

Table 72: Register overview

7.10.4.1 TASKS_STARTECB

Address offset: 0x000

Start ECB block encrypt

If a crypto operation is already running in the AES core, the STARTECB task will not start a new encryption and an ERRORECB event will be triggered

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTECB			Start ECB block encrypt If a crypto operation is already running in the AES core, the STARTECB task will not start a new encryption and an ERRORECB event will be triggered
			Trigger	1	Trigger task

7.10.4.2 TASKS_STOPECB

Address offset: 0x004

Abort a possible executing ECB operation

If a running ECB operation is aborted by STOPECB, the ERRORECB event is triggered.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOPECB			Abort a possible executing ECB operation If a running ECB operation is aborted by STOPECB, the ERRORECB event is triggered.
			Trigger	1	Trigger task

7.10.4.3 SUBSCRIBE_STARTECB

Address offset: 0x080

Subscribe configuration for task STARTECB

If a crypto operation is already running in the AES core, the STARTECB task will not start a new encryption and an ERRORECB event will be 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				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTECB will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.10.4.4 SUBSCRIBE_STOPECB

Address offset: 0x084

Subscribe configuration for task STOPECB

If a running ECB operation is aborted by STOPECB, the ERRORECB event is triggered.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOPECB will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.10.4.5 EVENTS_ENDECB

Address offset: 0x100

ECB block encrypt complete

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDECB			ECB block encrypt complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.10.4.6 EVENTS_ERRORECB

Address offset: 0x104

ECB block encrypt aborted because of a STOPECB task or due to an error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ERRORECB			ECB block encrypt aborted because of a STOPECB task or due to an error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.10.4.7 PUBLISH_ENDECB

Address offset: 0x180

Publish configuration for event ENDECB

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDECB will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.10.4.8 PUBLISH_ERRORECB

Address offset: 0x184

Publish configuration for event ERRORECB

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERRORECB will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.10.4.9 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENDECB			Write '1' to enable interrupt for event ENDECB
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERRORECB			Write '1' to enable interrupt for event ERRORECB
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.10.4.10 INTENCLR

Address offset: 0x308

Disable interrupt

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

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

7.10.4.11 ECBDATAPTR

Address offset: 0x504

ECB block encrypt memory pointers

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

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

7.10.5 Electrical specification

7.10.5.1 ECB Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{ECB}	Run time per 16 byte block in all modes			6.2	µs

7.11 EGU — Event generator unit

Event generator unit (EGU) provides support for interlayer signaling. This means providing 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 interrupt service routine (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:

- Software-enabled interrupt triggering
- Separate interrupt vectors for every EGU instance
- Up to 16 separate event flags per interrupt for multiplexing

Each instance of EGU implements a set of tasks which can individually be triggered to generate the corresponding event, for example, the corresponding event for TASKS_TRIGGER[n] is EVENTS_TRIGGERED[n]. See Instances on page 212 for a list of EGU instances.

7.11.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5001B000 0x4001B000	APPLICATION	EGU	EGU0 : S EGU0 : NS	US	NA	Event generator unit 0	
0x5001C000 0x4001C000	APPLICATION	EGU	EGU1 : S EGU1 : NS	US	NA	Event generator unit 1	
0x5001D000 0x4001D000	APPLICATION	EGU	EGU2 : S EGU2 : NS	US	NA	Event generator unit 2	
0x5001E000 0x4001E000	APPLICATION	EGU	EGU3 : S EGU3 : NS	US	NA	Event generator unit 3	
0x5001F000 0x4001F000	APPLICATION	EGU	EGU4 : S EGU4 : NS	US	NA	Event generator unit 4	
0x50020000 0x40020000	APPLICATION	EGU	EGU5 : S EGU5 : NS	US	NA	Event generator unit 5	
0x41014000	NETWORK	EGU	EGU0	NS	NA	Event generator unit 0	

Table 73: Instances

Register	Offset	Security	Description
TASKS_TRIGGER[n]	0x000		Trigger n for triggering the corresponding TRIGGERED[n] event
SUBSCRIBE_TRIGGER[n]	0x080		Subscribe configuration for task TRIGGER[n]
EVENTS_TRIGGERED[n]	0x100		Event number n generated by triggering the corresponding TRIGGER[n] task
PUBLISH_TRIGGERED[n]	0x180		Publish configuration for event TRIGGERED[n]
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt

Table 74: Register overview

7.11.1.1 TASKS_TRIGGER[n] (n=0..15)

Address offset: 0x000 + (n × 0x4)

Trigger n for triggering the corresponding TRIGGERED[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	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_TRIGGER			Trigger n for triggering the corresponding TRIGGERED[n] event
			Trigger	1	Trigger task

7.11.1.2 SUBSCRIBE_TRIGGER[n] (n=0..15)

Address offset: 0x080 + (n × 0x4)

Subscribe configuration for task TRIGGER[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task TRIGGER[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.11.1.3 EVENTS_TRIGGERED[n] (n=0..15)

Address offset: 0x100 + (n × 0x4)

Event number n generated by triggering the corresponding TRIGGER[n] 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_TRIGGERED			Event number n generated by triggering the corresponding TRIGGER[n] task
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.11.1.4 PUBLISH_TRIGGERED[n] (n=0..15)

Address offset: 0x180 + (n × 0x4)

Publish configuration for event TRIGGERED[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TRIGGERED[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.11.1.5 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A-P	RW	TRIGGERED[i] (i=0..15)			Enable or disable interrupt for event TRIGGERED[i]
			Disabled	0	Disable
			Enabled	1	Enable

7.11.1.6 INTENSET

Address offset: 0x304

Enable interrupt

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

7.11.1.7 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.11.2 Electrical specification

7.11.2.1 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

7.12 FPU - Floating point unit (FPU) exceptions

The Arm Cortex-M33 has FPU signals that indicate mathematical errors that cause floating-point exceptions.

The FPU signals are routed to the following event registers:

- FPUIOC: INVALIDOPERATION
- FPUIDC: DENORMALINPUT
- FPUOFC: OVERFLOW
- FPUUFC: UNDERFLOW
- FPUDZC: DIVIDEBYZERO
- FPUIXC: INEXACT

To clear the FPU exception source, write a 0 to the Arm Cortex-M33 FPSCR (floating-point status control register), as described in the *Arm Cortex-M33 Devices Generic User Guide*.

7.12.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50000000 0x40000000	APPLICATION	FPU	FPU : S FPU : NS	US	NA	Floating Point unit interrupt control	

Table 75: Instances

Register	Offset	Security	Description
EVENTS_INVALIDOPERATION	0x100		An FPUIOC exception triggered by an invalid operation has occurred in the FPU
EVENTS_DIVIDEBYZERO	0x104		An FPUDZC exception triggered by a floating-point divide-by-zero operation has occurred in the FPU
EVENTS_OVERFLOW	0x108		An FPUOFC exception triggered by a floating-point overflow has occurred in the FPU
EVENTS_UNDERFLOW	0x10C		An FPUUFC exception triggered by a floating-point underflow has occurred in the FPU
EVENTS_INEXACT	0x110		An FPUIXC exception triggered by an inexact floating-point operation has occurred in the FPU
EVENTS_DENORMALINPUT	0x114		An FPUIDC exception triggered by a denormal floating-point input has occurred in the FPU
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt

Table 76: Register overview

7.12.1.1 EVENTS_INVALIDOPERATION

Address offset: 0x100

An FPUIOC exception triggered by an invalid operation has occurred in the FPU

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_INVALIDOPERATION			An FPUIOC exception triggered by an invalid operation has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.2 EVENTS_DIVIDEBYZERO

Address offset: 0x104

An FPUDZC exception triggered by a floating-point divide-by-zero operation has occurred in the FPU

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DIVIDEBYZERO			An FPUDZC exception triggered by a floating-point divide-by-zero operation has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.3 EVENTS_OVERFLOW

Address offset: 0x108

An FPUOFC exception triggered by a floating-point overflow has occurred in the FPU

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_OVERFLOW			An FPUOFC exception triggered by a floating-point overflow has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.4 EVENTS_UNDERFLOW

Address offset: 0x10C

An FPUUFC exception triggered by a floating-point underflow has occurred in the FPU

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_UNDERFLOW			An FPUUFC exception triggered by a floating-point underflow has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.5 EVENTS_INEXACT

Address offset: 0x110

An FPUIXC exception triggered by an inexact floating-point operation has occurred in the FPU

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_INEXACT			An FPUIXC exception triggered by an inexact floating-point operation has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.6 EVENTS_DENORMALINPUT

Address offset: 0x114

An FPUIDC exception triggered by a denormal floating-point input has occurred in the FPU

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DENORMALINPUT			An FPUIDC exception triggered by a denormal floating-point input has occurred in the FPU
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.12.1.7 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 E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	INVALIDOPERATION			Enable or disable interrupt for event INVALIDOPERATION
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	DIVIDEBYZERO			Enable or disable interrupt for event DIVIDEBYZERO
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	OVERFLOW			Enable or disable interrupt for event OVERFLOW
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	UNDERFLOW			Enable or disable interrupt for event UNDERFLOW
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	INEXACT			Enable or disable interrupt for event INEXACT
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	DENORMALINPUT			Enable or disable interrupt for event DENORMALINPUT
			Disabled	0	Disable
			Enabled	1	Enable

7.12.1.8 INTENSET

Address offset: 0x304

Enable interrupt

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

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

ID	R/W	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DIVIDEBYZERO			Write '1' to enable interrupt for event DIVIDEBYZERO
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	OVERFLOW			Write '1' to enable interrupt for event OVERFLOW
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	UNDERFLOW			Write '1' to enable interrupt for event UNDERFLOW
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	INEXACT			Write '1' to enable interrupt for event INEXACT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	DENORMALINPUT			Write '1' to enable interrupt for event DENORMALINPUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.12.1.9 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	INVALIDOPERATION			Write '1' to disable interrupt for event INVALIDOPERATION
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DIVIDEBYZERO			Write '1' to disable interrupt for event DIVIDEBYZERO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	OVERFLOW			Write '1' to disable interrupt for event OVERFLOW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	UNDERFLOW			Write '1' to disable interrupt for event UNDERFLOW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

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

ID	R/W	Field	Value ID	Value	Description
E	RW	INEXACT			Write '1' to disable interrupt for event INEXACT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	DENORMALINPUT			Write '1' to disable interrupt for event DENORMALINPUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.13 GPIO — General purpose input/output

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

The number of ports and GPIOs per port may vary with product variant and package. Refer to Registers on page 225 and Pin assignments on page 783 for more information about the number of GPIOs that are supported.

GPIO has the following user-configurable features:

- Up to 32 GPIO pins per GPIO port
- Configurable output drive strength
- Internal pull-up and pull-down resistors
- Wake-up from high or low level triggers on all pins
- Trigger interrupt on state changes on any pin
- All pins can be used by the PPI task/event system
- One or more GPIO outputs can be controlled through PPI and GPIOTE channels
- All pins can be individually mapped to interface blocks for layout flexibility
- GPIO state changes captured on SENSE signal can be stored by LATCH register
- Pin sharing in multi-MCU system
- Support for secure and non-secure attributes for pins in conjunction with the system protection unit (SPU — System protection unit on page 584)

GPIO port and the GPIO pin details on page 220 illustrates the GPIO port containing 32 individual pins, where `PIN0` is illustrated in more detail as a reference. All signals on the left side in the illustration are used by other peripherals in the system and therefore not directly available to the CPU.

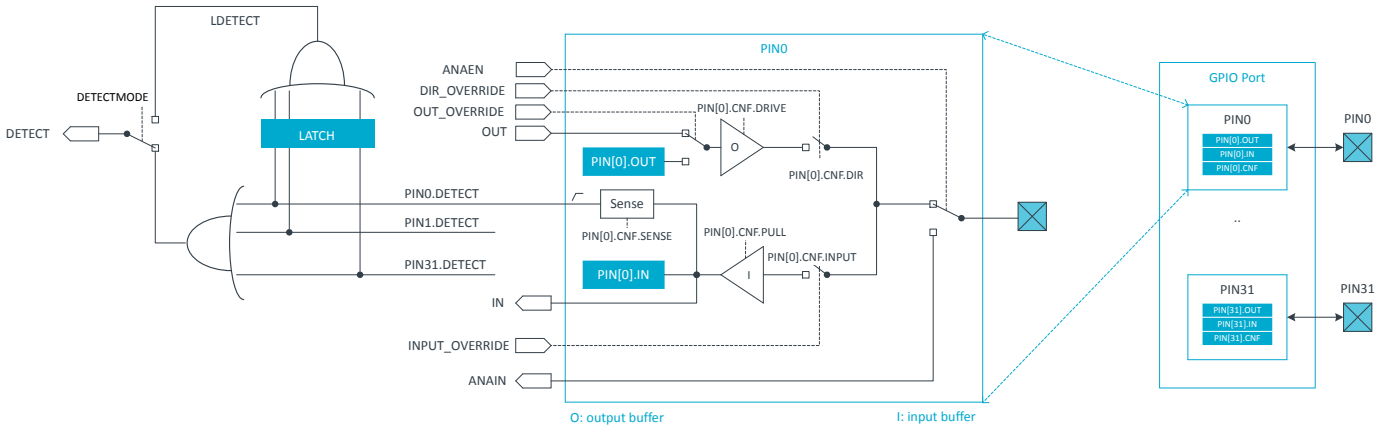

Figure 57: GPIO port and the GPIO pin details

7.13.1 Assigning pins between cores, peripherals, or subsystems

GPIO pins of the system can be allocated among the cores, peripherals with dedicated pins, or subsystems such as trace and debug.

The pins of the system are listed in Pin assignments on page 783.

A pin can be assigned to any of the following:

- Application core
- Network core
- Peripheral with dedicated pins
- Trace and debug (TaD) subsystem

By default, all pins are assigned to the application core. The application core's MCUSEL bitfield in register PIN_CNF[n] (n=0..31) (Retained) on page 228 controls the allocation of the pins into cores, peripherals and the TaD subsystem.

The pin's state, being either an output, input, or analog, can only be controlled and observed by the core for which the pin was allocated. Reading a pin that is not allocated to the current core will return zero, and writes will be ignored. If a pin is allocated to a subsystem that cannot access it, the pin stays under control of the application core's GPIO peripheral.

The GPIO peripheral of any core has its own set of registers that can be read and written independently, but they only affect a pin when it has been allocated to this core. Reading the GPIO peripheral registers of one core does not reveal the register contents of a different core.

The following figure illustrates how to assign a pin to a core, to a peripheral that has dedicated pins, or a subsystem such as trace and debug.

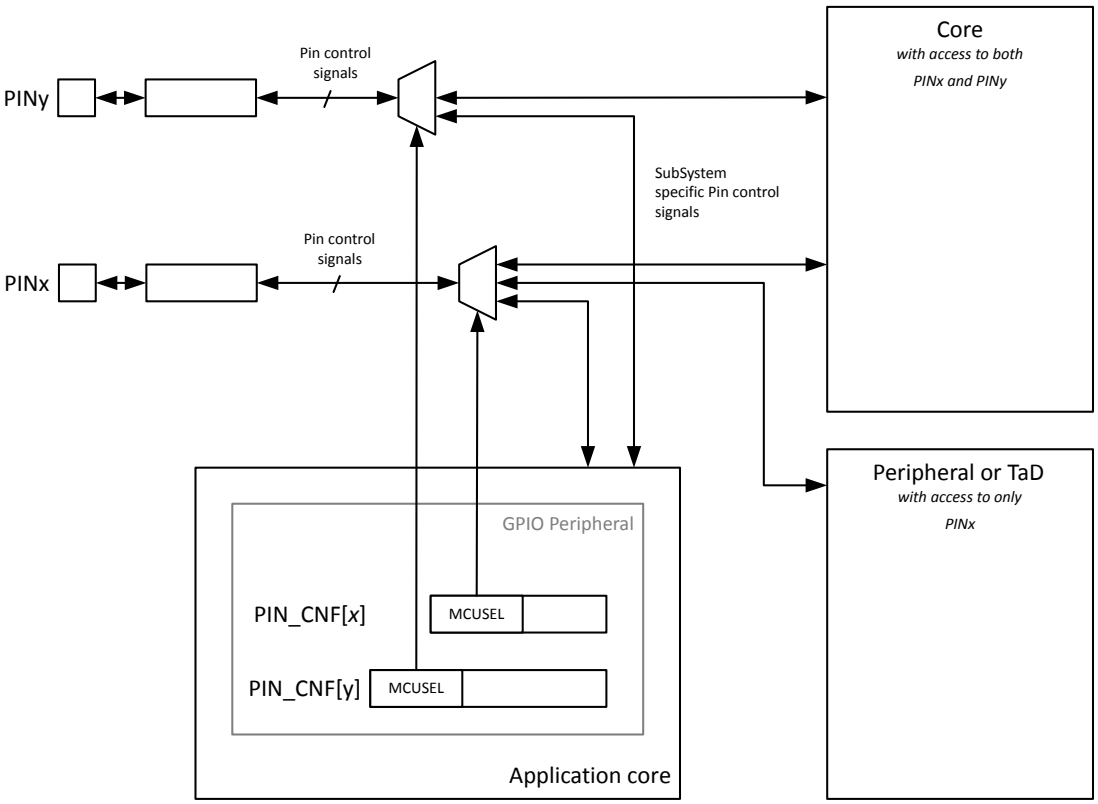

Figure 58: Pin allocation in multiple-core system

Note: To avoid glitches, changing the MCUSEL bitfield for a pin should only occur when the pin is disabled.

Also note that when a pin *p* is not assigned to the application core, the application core's GPIO LATCH register, PIN_CNF[*p*].MCUSEL bitfield, and PIN_CNF[*p*].SENSE bitfield is prevented. For these, any write operations are ignored, and any read operation will return 0.

7.13.2 Pin configuration

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).

Note: For more information on pin assignment and the corresponding effect of read and write operations of GPIO registers, see Assigning pins between cores, peripherals, or subsystems on page 220.

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)

Note: All write-capable registers are retained registers, see POWER — Power control on page 43 for more information.

The input buffer of a GPIO pin can be disconnected from the pin to enable power savings when the pin is not used as an input, see GPIO port and the GPIO pin details on page 220. Inputs must be connected to get a valid input value in the IN register, and for the sense mechanism to get access to the pin.

Other peripherals in the system can connect to GPIO pins and override their output value and configuration, or read their analog or digital input value. See GPIO port and the GPIO pin details on page 220.

Selected pins also support analog input signals, see ANAIN in GPIO port and the GPIO pin details on page 220. The assignment of the analog pins can be found in Pin assignments on page 783.

The drive strength is configured using the DRIVE field of register PIN_CNF[n] (n=0..31) (Retained) on page 228. Some pins may not support every drive configuration, see Pin assignments on page 783 for more information.

The following delays should be taken into considerations:

- There is a delay of 2 CPU clock cycles from the GPIO pad to the IN register.
- The GPIO pad must be low (or high depending on the SENSE polarity) for 3 CPU clock cycles after DETECT has gone high to generate a new DETECT signal.

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

7.13.3 Pin sense mechanism

Pins sensitivity 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.

Note: Refer to Assigning pins between cores, peripherals, or subsystems on page 220 for pin assignment and corresponding effect of read and write operations of GPIO registers

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

DETECTMODE and DETECTMODE_SEC are provided to handle secure and non-secure pins. DETECTMODE_SEC register is available to control the behavior associated to pin marked as secure, while the DETECTMODE register is restricted to pin marked as non-secure. Please refer to GPIO security on page 223 for more details.

Make sure that a pin is in a level that cannot trigger the sense mechanism before enabling it. The DETECT signal will go high immediately if the SENSE condition configured in the PIN_CNF registers is met when the sense mechanism is enabled. This will trigger a PORT event if the DETECT signal was low before enabling the sense mechanism.

The DETECT signal is also used by power and clock management system to exit from System OFF mode, and by GPIOTE to generate the PORT event. In addition GPIOTE_SEC is used for PORT event related to secure pins). See POWER — Power control on page 43 and GPIOTE — GPIO tasks and events on page 230 for more information about how the DETECT signal is used.

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

writing a '1' to the bit that shall be cleared, i.e. the LATCH register will not be affected by a PINx.DETECT signal being set low.

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

If one or more bits in the LATCH register are '1' after the CPU has performed a clear operation on the LATCH registers, a rising edge will be generated on the LDETECT signal. This is illustrated in DETECT signal behavior on page 223.

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

The LDETECT signal is by default not connected to the GPIO port's DETECT signal, but via the DETECTMODE register it is possible to change from default behavior to DETECT signal being derived directly from the LDETECT signal instead. See GPIO port and the GPIO pin details on page 220. DETECT signal behavior on page 223 illustrates the DETECT signal behavior for these two alternatives.

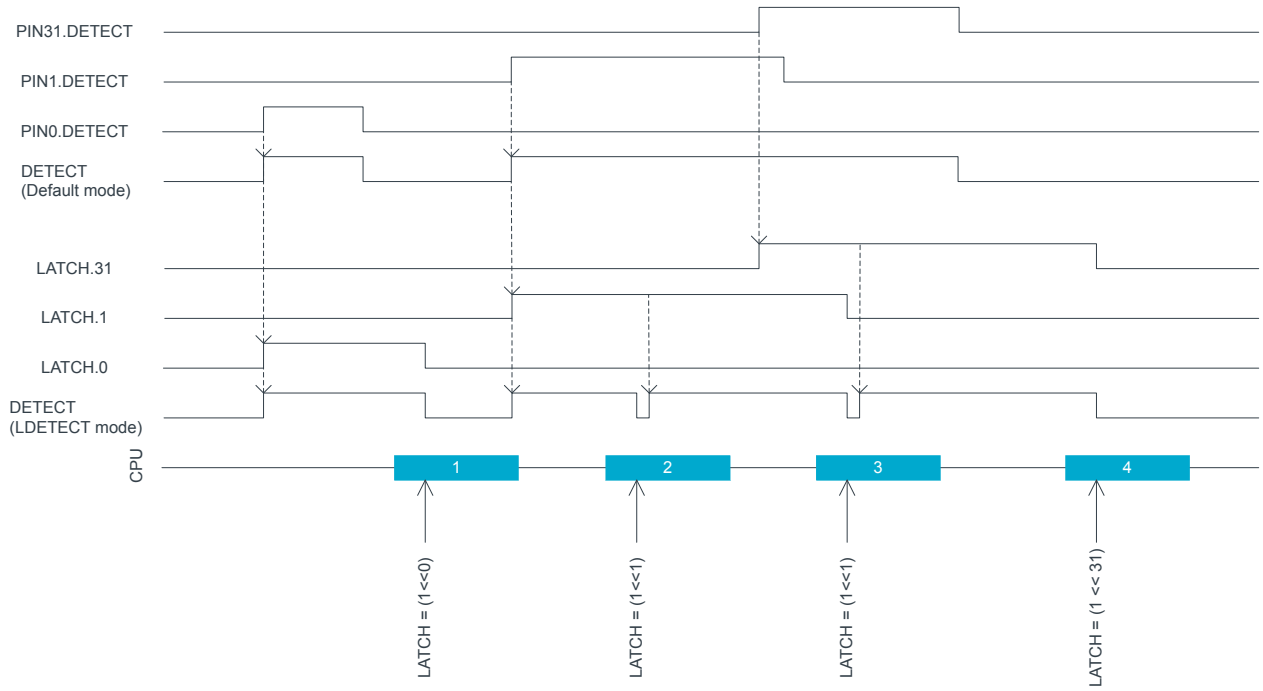

Figure 59: DETECT signal behavior

7.13.4 GPIO security

The general purpose input/output (GPIO) peripheral is implemented as a *split-security* peripheral. If marked as non-secure, it can be accessed by both secure and non-secure accesses but will behave differently depending on the access type.

Note: For more information on pin assignment and the corresponding effect of read and write operations of GPIO registers, see Assigning pins between cores, peripherals, or subsystems on page 220.

A non-secure peripheral access will only be able to configure and control pins defined as non-secure in the system protection unit (SPU) GPIOPORT.PERM[] register(s).

A non-secure access to a register or a bitfield controlling a pin marked as secure in GPIO.PERM[] register(s) will be ignored. Write access will have no effect and read access will return a zero value.

No exception is triggered when a non-secure access targets a register or bitfield controlling a secure pin. For example, if the bit `i` is set in the SPU.GPIO.PERM[0] register (declaring Pin P0.`i` as secure), then

- non-secure write accesses to OUT, OUTSET, OUTCLR, DIR, DIRSET, DIRCLR and LATCH registers will not be able to write to bit `i` of those registers
- non-secure write accesses to registers PIN[`i`].OUT and PIN_CNF[`i`] will be ignored
- non-secure read accesses to registers OUT, OUTSET, OUTCLR, IN, DIR, DIRSET, DIRCLR and LATCH will always read a '0' for the bit at position `i`
- non-secure read accesses to registers PIN[`i`].OUT, PIN[`i`].OUT and PIN_CNF[`i`] will always return 0

The GPIO.DETECTMODE and GPIO.DETECTMODE_SEC registers are handled differently than the other registers mentioned before. When accessed by a secure access, the DETECTMODE_SEC register control the source for the DETECT_SEC signal for the pins marked as secure. When accessed by a non-secure access, the DETECTMODE_SEC is read as zero and write accesses are ignored. The GPIO.DETECTMODE register controls the source for the DETECT_NSEC signal for the pins defined as non-secure.

The DETECT_NSEC signal is routed to the GPIOTE peripheral, allowing generation of events and interrupts from pins marked as non-secure. The DETECT_SEC signal is routed to the GPIOTESEC peripheral, allowing generation of events and interrupts from pins marked as secure. Principle of direct pin access on page 224 illustrates how the DETECT_NSEC and DETECT_SEC signals are generated from the GPIO PIN[].DETECT signals.

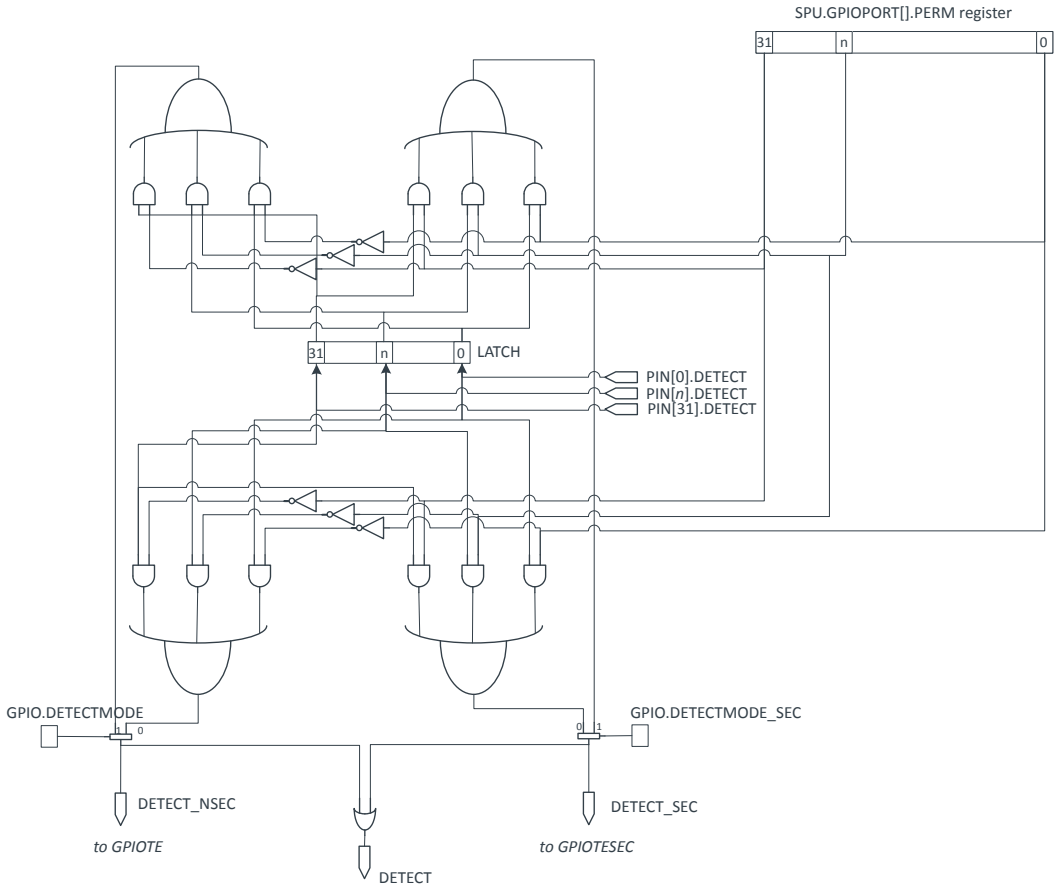

Figure 60: Principle of direct pin access

7.13.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50842500 0x40842500	APPLICATION	GPIO	P0 : S P0 : NS	SPLIT	NA	General purpose input and output, port 0	P0.00 to P0.31 implemented
0x50842800 0x40842800	APPLICATION	GPIO	P1 : S P1 : NS	SPLIT	NA	General purpose input and output, port 1	P1.00 to P1.15 implemented
0x418C0500	NETWORK	GPIO	P0	NS	NA	General purpose input and output	P0.00 to P0.31 implemented
0x418C0800	NETWORK	GPIO	P1	NS	NA	General purpose input and output	P1.00 to P1.15 implemented

Table 77: Instances

Register	Offset	Security	Description	
OUT	0x004		Write GPIO port	Retained
OUTSET	0x008		Set individual bits in GPIO port	
OUTCLR	0x00C		Clear individual bits in GPIO port	
IN	0x010		Read GPIO port	
DIR	0x014		Direction of GPIO pins	Retained
DIRSET	0x018		DIR set register	
DIRCLR	0x01C		DIR clear register	
LATCH	0x020		Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers	Retained
DETECTMODE	0x024		Select between default DETECT signal behavior and LDETECT mode (For non-secure pin only)	Retained
DETECTMODE_SEC	0x028		Select between default DETECT signal behavior and LDETECT mode (For secure pin only)	Retained
PIN_CNF[n]	0x200		Configuration of GPIO pins	Retained

Table 78: Register overview

7.13.5.1 OUT (Retained)

Address offset: 0x004

This register is a retained register

Write GPIO port

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

ID	R/W	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Pin i
			Low	0	Pin driver is low
			High	1	Pin driver is high

7.13.5.2 OUTSET

Address offset: 0x008

Set individual bits in GPIO port

> **Note:** Read: reads value of OUT register.

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

7.13.5.3 OUTCLR

Address offset: 0x00C

Clear individual bits in GPIO port

Note: Read: reads value of OUT register.

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

7.13.5.4 IN

Address offset: 0x010

Read GPIO port

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

7.13.5.5 DIR (Retained)

Address offset: 0x014

This register is a retained register

Direction of GPIO pins

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

7.13.5.6 DIRSET

Address offset: 0x018

DIR set register

Note: Read: reads value of DIR register.

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

7.13.5.7 DIRCLR

Address offset: 0x01C

DIR clear register

Note: Read: reads value of DIR register.

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

7.13.5.8 LATCH (Retained)

Address offset: 0x020

This register is a retained register

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

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

7.13.5.9 DETECTMODE (Retained)

Address offset: 0x024

This register is a retained register

Select between default DETECT signal behavior and LDETECT mode (For non-secure pin only)

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

7.13.5.10 DETECTMODE_SEC (Retained)

Address offset: 0x028

This register is a retained register

Select between default DETECT signal behavior and LDETECT mode (For secure pin only)

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

7.13.5.11 PIN_CNF[n] (n=0..31) (Retained)

Address offset: 0x200 + (n × 0x4)

This register is a retained register

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				G G G E E D D D D C C B A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DIR			Pin direction. Same physical register as DIR register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G G G E E D D D D C C B A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
			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 **Note:** Some pins may not support every 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)
			E0E1	11	Extra high drive '0', extra high drive '1'
E	RW	SENSE			Pin sensing mechanism
			Disabled	0	Disabled
			High	2	Sense for high level
			Low	3	Sense for low level
G	RW	MCUSEL			Select which MCU/Subsystem controls this pin Note: this field is only accessible from secure code.
			AppMCU	0x0	Application MCU
			NetworkMCU	0x1	Network MCU
			Peripheral	0x3	Peripheral with dedicated pins
			TND	0x7	Trace and Debug Subsystem

7.13.6 Electrical specification

7.13.6.1 GPIO Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage	0.7 x VDD		VDD	V
V_{IL}	Input low voltage	VSS		0.3 x VDD	V
$V_{OH,SD}$	Output high voltage, standard drive, 0.5 mA, VDD ≥ 1.7 V	VDD - 0.4		VDD	V

Symbol	Description	Min.	Typ.	Max.	Units
$V_{OH,HDH}$	Output high voltage, high drive, 5 mA, VDD ≥ 2.7 V	VDD - 0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 3 mA, VDD ≥ 1.7 V	VDD - 0.4		VDD	V
$V_{OL,SD}$	Output low voltage, standard drive, 0.5 mA, VDD ≥ 1.7 V	VSS		VSS + 0.4	V
$V_{OL,HDH}$	Output low voltage, high drive, 5 mA, VDD ≥ 2.7 V	VSS		VSS + 0.4	V
$V_{OL,HDL}$	Output low voltage, high drive, 3 mA, VDD ≥ 1.7 V	VSS		VSS + 0.4	V
$I_{OL,SD}$	Current at VSS + 0.4 V, output set low, standard drive, VDD ≥ 1.7 V	1	2	4	mA
$I_{OL,HDH}$	Current at VSS + 0.4 V, output set low, high drive, VDD ≥ 2.7 V	6			mA
$I_{OL,HDL}$	Current at VSS + 0.4 V, output set low, high drive, VDD ≥ 1.7 V	3			mA
$I_{OL,HDL,QSPI}$	Current at VSS + 0.4 V, output set low, high drive, VDD ≥ 1.7 V		10		mA
$I_{OL,HDL,TWIM}$	Current at VSS + 0.4 V, output set low, high drive, VDD ≥ 1.7 V		50		mA
$I_{OH,SD}$	Current at VDD - 0.4 V, output set high, standard drive, VDD ≥1.7	1	2	3	mA
$I_{OH,HDH}$	Current at VDD - 0.4 V, output set high, high drive, VDD ≥ 2.7 V	6			mA
$I_{OH,HDL}$	Current at VDD - 0.4 V, output set high, high drive, VDD ≥ 1.7 V	3			mA
$I_{OH,HDL,QSPI}$	Current at VDD - 0.4 V, output set high, high drive, VDD ≥ 1.7 V		10		mA
$t_{RF,15pF}$	Rise/fall time, standard drive mode, 10 to 90%, 15 pF load[1]		9		ns
$t_{RF,25pF}$	Rise/fall time, standard drive mode, 10 to 90%, 25 pF load[1]		14		ns
$t_{RF,50pF}$	Rise/fall time, standard drive mode, 10 to 90%, 50 pF load[1]		26		ns
$t_{HRF,10pF,QSPI96}$	Rise/Fall time, high drive mode, 20 to 80%, 10 pF load, VDD 1.6 V to 3.6 V, QSPI running at 96 MHz[1]		8.5		ns
$t_{HRF,15pF}$	Rise/Fall time, high drive mode, 10 to 90%, 15 pF load[1]		4		ns
$t_{HRF,25pF}$	Rise/Fall time, high drive mode, 10 to 90%, 25 pF load[1]		5		ns
$t_{HRF,50pF}$	Rise/Fall time, high drive mode, 10 to 90%, 50 pF load[1]		9		ns
R_{PU}	Pull-up resistance		13		kΩ
R_{PD}	Pull-down resistance		13		kΩ
C_{PAD}	Pad capacitance		1.5		pF
C_{PAD_NFC}	Pad capacitance on NFC pads		4		pF
I_{NFC_LEAK}	Leakage current between NFC pads when driven to different states		1	10	µA

7.14 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. Tasks and events are briefly introduced in Peripheral interface on page 145, and GPIO is described in more detail in GPIO — General purpose input/output on page 219.

[1] Rise and fall times based on simulations

Low power detection of pin state changes is possible when in System ON or System OFF.

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

7.14.1 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 secure instance of the GPIOTE peripheral is able to operate on all GPIO pins configured in GPIOTE.CONFIG[n].PSEL.

The non-secure instance of the GPIOTE peripheral is able to operate only on non-secure GPIO pins. The field GPIOTE.CONFIG[n].PSEL can only select a non-secure pin.

The tasks SET[n], CLR[n], and OUT[n] can write to individual pins, and events IN[n] can be generated from input changes of individual pins.

The SET task will set the pin selected in GPIOTE.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. It can set the pin high, set it low, or toggle it.

Tasks and events are configured using the CONFIG[n] registers. One CONFIG[n] register is associated with a set of SET[n], CLR[n], and OUT[n] tasks and IN[n] events.

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

When conflicting tasks are triggered simultaneously (i.e. during the same clock cycle) in one channel, the priority of the tasks is as described in the following table.

Priority	Task
1	OUT
2	CLR
3	SET>

Table 79: Task priorities

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

7.14.2 Port event

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

The event will be generated on the rising edge of the DETECT signal. See GPIO — General purpose input/output on page 219 for more information about the DETECT signal.

There are two DETECT signals that come from the GPIO peripheral. The secure DETECT_SEC, for the secure instance of the GPIOTE peripheral, and the non-secure DETECT_NONSEC, for the non-secure instance of the GPIOTE peripheral.

The GPIO DETECT signal will not wake the system up again if the system is put into System ON IDLE while the DETECT signal is high. Clear all DETECT sources before entering sleep. If the LATCH register is used as a source, a new rising edge will be generated on DETECT if any bit in LATCH is still high after clearing all or part of the register. This could occur if one of the PINx.DETECT signals is still high, for example. See Pin sense mechanism on page 222 for more information.

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

This feature can be used to wake up the CPU from a WFI or WFE type sleep in System ON when all peripherals and the CPU are idle, meaning the lowest power consumption in System ON mode.

To prevent spurious interrupts from the PORT event while configuring the sources, the following steps must be performed:

1. Disable interrupts on the PORT event (through INTENCLR.PORT).
2. Configure the sources (PIN_CNF[n].SENSE).
3. Clear any potential event that could have occurred during configuration (write `0` to EVENTS_PORT).
4. Enable interrupts (through INTENSET.PORT).

7.14.3 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.

Note: A pin can only be assigned to one GPIOTE channel at a time. Failing to do so may result in unpredictable behavior.

7.14.4 Low power

In Event mode (CONFIG.MODE=`Event`), the register LATENCY on page 237 makes it possible to trade off detection speed (latency) versus power consumption.

If the input signal on the GPIO pin has sufficient hold time, LATENCY=`LowPower` can be used, thus saving power. Figure below shows hold times for `LoToHi` transitions, but the same conditions apply for `HiToLo`.

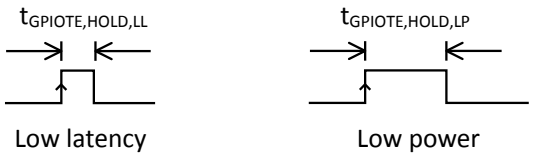

Figure 61: GPIOTE minimum hold time for different GPIOTE.LATENCY selections

For the values of minimum hold times, refer to .

LowPower mode can be used with both PORT event (EVENTS_PORT) and IN event (EVENTS_IN). The GPIO pins must have their PIN_CNF[n].SENSE configured.

When using LowPower with IN event, the GPIO PIN_CNF[] register's SENSE field needs to be configured according to CONFIG.POLARITY for the selected pin.

- When CONFIG.POLARITY=`LoToHi`, SENSE must be set to `High`.
- When CONFIG.POLARITY=`HiToLo`, SENSE must be set to `Low`.

When using LowPower with PORT event, the GPIO PIN_CNF[] register's SENSE field can have any value.

CONFIG.POLARITY=`Toggle` is not supported for LATENCY=`LowPower`.

7.14.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5000D000	APPLICATION	GPIOTE	GPIOTE0	S	NA	GPIO tasks and events	
0x4002F000	APPLICATION	GPIOTE	GPIOTE1	NS	NA	GPIO tasks and events	
0x4100A000	NETWORK	GPIOTE	GPIOTE	NS	NA	GPIO tasks and events	

Table 80: Instances

Register	Offset	Security	Description
TASKS_OUT[n]	0x000		Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is configured in CONFIG[n].POLARITY.
TASKS_SET[n]	0x030		Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it high.
TASKS_CLR[n]	0x060		Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it low.
SUBSCRIBE_OUT[n]	0x080		Subscribe configuration for task OUT[n]
SUBSCRIBE_SET[n]	0x0B0		Subscribe configuration for task SET[n]
SUBSCRIBE_CLR[n]	0x0E0		Subscribe configuration for task CLR[n]
EVENTS_IN[n]	0x100		Event generated from pin specified in CONFIG[n].PSEL
EVENTS_PORT	0x17C		Event generated from multiple input GPIO pins with SENSE mechanism enabled
PUBLISH_IN[n]	0x180		Publish configuration for event IN[n]
PUBLISH_PORT	0x1FC		Publish configuration for event PORT
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
LATENCY	0x504		Latency selection for Event mode (MODE=Event) with rising or falling edge detection on the pin.
CONFIG[n]	0x510		Configuration for OUT[n], SET[n], and CLR[n] tasks and IN[n] event

Table 81: Register overview

7.14.5.1 TASKS_OUT[n] (n=0..7)

Address offset: 0x000 + (n × 0x4)

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is configured in CONFIG[n].POLARITY.

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_OUT			Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is configured in CONFIG[n].POLARITY.
			Trigger	1	Trigger task

7.14.5.2 TASKS_SET[n] (n=0..7)

Address offset: 0x030 + (n × 0x4)

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it high.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_SET			Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it high.
			Trigger	1	Trigger task

7.14.5.3 TASKS_CLR[n] (n=0..7)

Address offset: 0x060 + (n × 0x4)

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_CLR			Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it low.
			Trigger	1	Trigger task

7.14.5.4 SUBSCRIBE_OUT[n] (n=0..7)

Address offset: 0x080 + (n × 0x4)

Subscribe configuration for task OUT[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task OUT[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.14.5.5 SUBSCRIBE_SET[n] (n=0..7)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task SET[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SET[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.14.5.6 SUBSCRIBE_CLR[n] (n=0..7)

Address offset: 0x0E0 + (n × 0x4)

Subscribe configuration for task CLR[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CLR[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.14.5.7 EVENTS_IN[n] (n=0..7)

Address offset: 0x100 + (n × 0x4)

Event generated from pin specified in CONFIG[n].PSEL

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_IN			Event generated from pin specified in CONFIG[n].PSEL
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.14.5.8 EVENTS_PORT

Address offset: 0x17C

Event generated from multiple input GPIO pins with SENSE mechanism enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_PORT			Event generated from multiple input GPIO pins with SENSE mechanism enabled
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.14.5.9 PUBLISH_IN[n] (n=0..7)

Address offset: 0x180 + (n × 0x4)

Publish configuration for event IN[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event IN[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.14.5.10 PUBLISH_PORT

Address offset: 0x1FC

Publish configuration for event 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event PORT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.14.5.11 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	IN[i] (i=0..7)			Write '1' to enable interrupt for event IN[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PORT			Write '1' to enable interrupt for event PORT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.14.5.12 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	IN[i] (i=0..7)			Write '1' to disable interrupt for event IN[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	PORT			Write '1' to disable interrupt for event PORT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.14.5.13 LATENCY

Address offset: 0x504

Latency selection for Event mode (MODE=Event) with rising or falling edge detection on the pin.

POLARITY=Toggle can only be used with LATENCY=LowLatency.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	LATENCY			Latency setting
			LowPower	0	Low power setting, for signals with minimum hold time tGPIOTE,HOLD,LP; refer to Electrical specification section
			LowLatency	1	Low latency setting, for signals with minimum hold time tGPIOTE,HOLD,LL; refer to Electrical specification section

7.14.5.14 CONFIG[n] (n=0..7)

Address offset: 0x510 + (n × 0x4)

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

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D D C B B B B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MODE			Mode
			Disabled	0	Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.
			Event	1	Event mode The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.

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

ID	R/W	Field	Value ID	Value	Description
			Task	3	Task mode The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.
B	RW	PSEL		[0..31]	GPIO number associated with SET[n], CLR[n], and OUT[n] tasks and IN[n] event
C	RW	PORT		[0..1]	Port number
D	RW	POLARITY			When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.
			None	0	Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.
			LoToHi	1	Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.
			HiToLo	2	Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.
			Toggle	3	Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin. POLARITY=Toggle requires the use of LATENCY=LowLatency.
E	RW	OUTINIT			When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.
			Low	0	Task mode: Initial value of pin before task triggering is low
			High	1	Task mode: Initial value of pin before task triggering is high

7.14.6 Electrical specification

7.15 I²S — Inter-IC sound interface

The I²S (Inter-IC Sound) module, supports the original two-channel I²S format, and left- or right-aligned formats. It implements EasyDMA for sample transfer directly to and from RAM without CPU intervention.

The I²S peripheral has the following main features:

- Master and Slave mode
- Simultaneous bidirectional (TX and RX) audio streaming
- Original I²S and left- or right-aligned format
- 32, 24, 16 and 8-bit sample widths
- Separate sample and word widths
- Low-jitter master clock generator
- Various sample rates

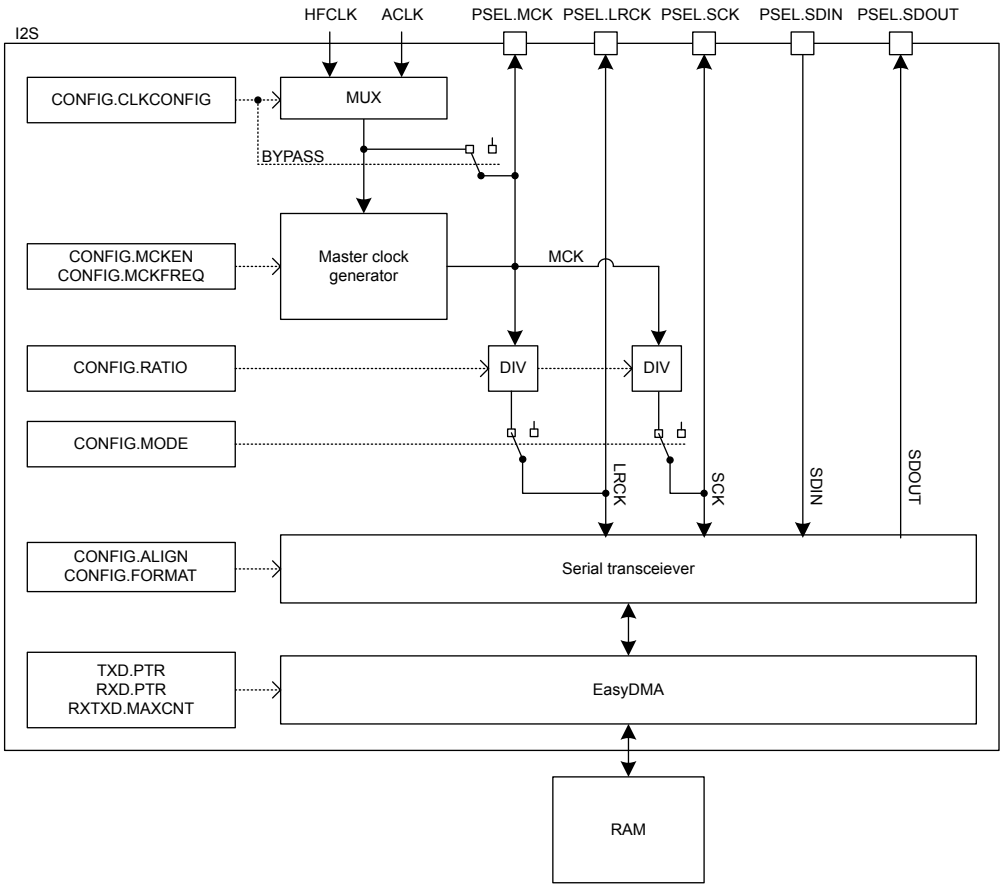

Figure 62: I²S master

7.15.1 Mode

The I²S protocol specification defines two modes of operation, Master and Slave.

The I²S mode decides which of the two sides (master or slave) shall provide the clock signals LRCK and SCK, and these signals are always supplied by the master to the slave.

7.15.2 Transmitting and receiving

The I²S module supports both transmission (TX) and reception (RX) of serial data. In both cases the serial data is shifted synchronously to the clock signals SCK and LRCK.

TX data is written to the SDOUT pin on the falling edge of SCK, and RX data is read from the SDIN pin on the rising edge of SCK. The most significant bit (MSB) is always transmitted first.

Note: When starting a transmission in master mode, the first frame is filled with zeros.

TX and RX are available in both Master and Slave modes and can be enabled/disabled independently in the CONFIG.TXEN on page 257 and CONFIG.RXEN on page 257.

Transmission and/or reception is started by triggering the START task. With transmission enabled in CONFIG.TXEN), the TXPTRUPD event will be generated for every number of transmitted data words given by RXTXD.MAXCNT on page 262. Each data word contains one or more samples. The TXPTRUPD event is generated just before MAXCNT number of data words have been transmitted. Similarly, with reception enabled in CONFIG.RXEN, the RXPTRUPD event will be generated for every received data word given by RXTXD.MAXCNT on page 262. The RXPTRUPD event is generated just after MAXCNT number of data words have been received.

The FRAMESTART event is generated synchronously to the active LRCK edge at the beginning of a frame after transmitting RXTXD.MAXCNT data words. The initial FRAMESTART event is generated at the first active edge of LRCK after the START task has been triggered. The FRAMESTART event is only defined for transmitting full left and right sample pairs. If MAXCNT is configured so that the frame ends between the left and right sample pairs, the FRAMESTART event is not generated. This occurs for the following combinations of SWIDTH and MAXCNT:

SWIDTH	MAXCNT restriction
24Bit	Only even numbers (2,4,6, etc)
32	Only even numbers (2, 4, 6, etc)
24BitIn32	Only even numbers (2, 4, 6, etc)

Table 82: Restrictions on combinations of SWIDTH and MAXCNT for correct FRAMESTART

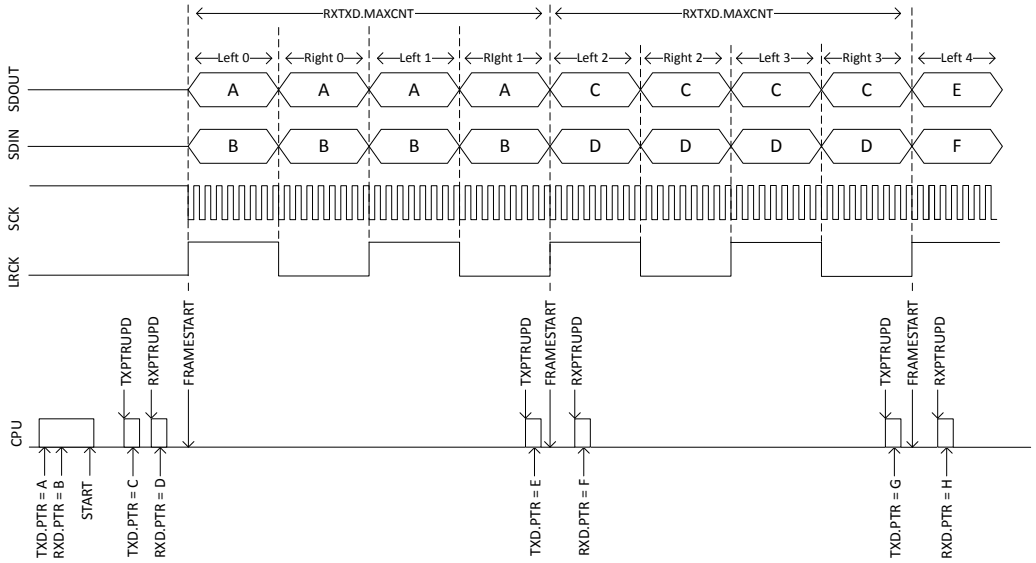

Figure 63: Transmitting and receiving. CONFIG.FORMAT = Aligned, CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo, RXTXD.MAXCNT = 1

7.15.3 Left right clock (LRCK)

The left right clock (LRCK), often referred to as word clock, sample clock, or word select in I^2S context, is the clock defining the frames in serial bitstreams sent and received on SDOUT and SDIN, respectively.

In I2S format, each frame contains one left and/or right sample pair. The left sample is transferred during the low half period of LRCK, followed by the right sample being transferred during the high half period of LRCK.

In Aligned format, each frame contains one left and/or right sample pair. The left sample is transferred during the high half period of LRCK, followed by the right sample being transferred during the low half period of LRCK.

For mono, the frame will contain only zeros for the unused half period of LRCK.

Consequently, the LRCK frequency is equivalent to the audio sample rate.

When operating in Master mode, the LRCK is generated from the MCK, and the frequency of LRCK is then given as:

```
LRCK = MCK / CONFIG.RATIO
```

LRCK always toggles around the falling edge of the serial clock SCK.

7.15.4 Serial clock (SCK)

The serial clock (SCK), often referred to as the serial bit clock, pulses once for each data bit being transferred on the serial data lines SDIN and SDOUT.

When operating in Master mode, the SCK is generated from the MCK, and the frequency of SCK is then given as:

```
SCK = 2 * LRCK * CONFIG.SWIDTH
```

The falling edge of the SCK falls on the toggling edge of LRCK.

When operating in Slave mode, SCK is provided by the external I^2S master.

7.15.5 Master clock (MCK)

The master clock (MCK) is the clock from which LRCK and SCK are derived when operating in Master mode.

The master clock 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.

MCK is generated from the clock source selected in the CONFIG.CLKCONFIG and CONFIG.MCKFREQ registers.

The following equation can be used to calculate the value of CONFIG.MCKFREQ for given MCK and clock source frequency:

$$MCKFREQ = 4096 \cdot \left\lfloor \frac{f_{MCK} \cdot 1048576}{f_{source} + \frac{f_{MCK}}{2}} \right\rfloor$$

Figure 64: MCK clock frequency equation

The parameter f_{MCK} is the requested MCK clock frequency in Hz, and f_{source} is the frequency of the selected clock source in Hz. Because of rounding errors, an accurate MCK clock may not be achievable. The equation does not take into account the maximum register value of CONFIG.MCKFREQ on page 258.

The actual MCK frequency can be calculated using the equation below.

$$f_{actual} = \frac{f_{source}}{\left\lfloor \frac{1048576 \cdot 4096}{MCKFREQ} \right\rfloor}$$

Figure 65: Actual MCK clock frequency

The clock error can be calculated using the equation below. The error *e* is the percentage difference from the requested f_{MCK} frequency.

$$e = 100 \cdot \frac{f_{actual} - f_{MCK}}{f_{MCK}} = 100 \cdot \frac{\frac{f_{source}}{\left\lfloor \frac{1048576 \cdot 4096}{MCKFREQ} \right\rfloor} - f_{MCK}}{f_{MCK}}$$

Figure 66: MCK frequency error equation

The master clock generator does not add any jitter to the clock source chosen.

The master clock generator is enabled/disabled using CONFIG.MCKEN on page 257, and the generator is started or stopped by the START or STOP tasks respectively.

The MCK frequency can be adjusted on-the-fly:

- For PCLK32M, by using MCKFREQ

- For ACLK, by adjusting the audio clock source, see CLOCK — Clock control on page 69.

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 259 and CONFIG.SWIDTH on page 259.

When configuring these registers, the user is responsible for fulfilling the following requirements:

1. The SCK frequency can never exceed the MCK frequency.
2. The MCK/LRCK ratio shall be a multiple of 2 * CONFIG.SWIDTH.

The MCK signal can be routed to an output pin (specified in PSEL.MCK) to supply external I^2S devices that require the MCK to be supplied from the outside.

When operating in Slave mode, the I^2S module does not use the MCK and the MCK generator does not need to be enabled.

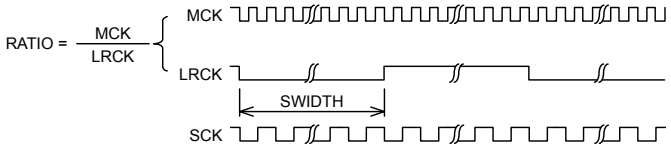

Figure 67: Relation between RATIO, MCK and LRCK

7.15.5.1 Clock source selection

The clock source for the master clock generator can be selected in the register CONFIG.CLKCONFIG on page 261. Choose one of the following clocks as the clock source:

- 32 MHz peripheral clock (PCLK32M), synchronous to HFCLK.
- Audio PLL clock (ACLK) with configurable frequency.

To improve the master clock accuracy and jitter performance, it is recommended (but not mandatory) that the PCLK32M source is running off the HFXO instead of the HFINT oscillator. The ACLK source requires the use of HFXO. See CLOCK — Clock control on page 69 for more information about starting HFXO for the relevant clock source.

The master clock generator can be bypassed so the MCK clock is derived directly from the input source. This can be configured in the BYPASS field of register CONFIG.CLKCONFIG on page 261.

7.15.5.2 Configuration examples

Configuration examples for CLKCONFIG = PCLK32M on page 243 and Configuration examples for CLKCONFIG = ACLK on page 243 show some configuration examples for popular sample rates, using both the 32 MHz master clock and the Audio PLL clock source.

Source frequency [Hz]	Requested LRCK [Hz]	RATIO	Requested MCK [Hz]	MCKFREQ	MCK [Hz]	LRCK [Hz]	LRCK error [%]
32000000	16000	32	512000	68173824	507936	15873	-0.8
32000000	16000	64	1024000	135274496	1032258	16129	0.8
32000000	16000	256	4096000	516685824	4000000	15625	-2.3
32000000	32000	32	1024000	135274496	1032258	32258	0.8
32000000	32000	64	2048000	266350592	2000000	31250	-2.3
32000000	32000	256	8192000	974741504	8000000	31250	-2.3
32000000	44100	32	1411200	185319424	1391304	43478	-1.4
32000000	44100	64	2822400	362815488	2909090	45455	3.1
32000000	48000	32	1536000	201326592	1523809	47619	-0.8
32000000	48000	64	3072000	393428992	3200000	50000	4.2
32000000	96000	32	3072000	393428992	3200000	100000	4.2
32000000	96000	64	6144000	752402432	6400000	100000	4.2

Table 83: Configuration examples for CLKCONFIG = PCLK32M

Source frequency [Hz]	Requested LRCK [Hz]	RATIO	Requested MCK [Hz]	MCKFREQ	MCK [Hz]	LRCK [Hz]	LRCK error [%]
11289600	44100	32	1411200	505286656	1411200	44100	0
11289600	44100	64	2822400	954433536	2822400	44100	0
12288000	16000	32	510000	175304704	512000	16000	0
12288000	16000	64	1024000	343597056	1024000	16000	0
12288000	32000	32	1024000	343597056	1024000	32000	0
12288000	32000	64	2048000	660762624	2048000	32000	0
12288000	48000	32	1536000	505286656	1536000	48000	0
12288000	48000	64	3072000	954433536	3072000	48000	0
12288000	96000	32	3072000	954433536	3072000	96000	0

Table 84: Configuration examples for CLKCONFIG = ACLK

7.15.6 Width, alignment and format

The register CONFIG.SWIDTH on page 259 defines the sample width of the data read and written to memory, as well as the number of SCK clock cycles per half-frame. Figure Aligned format, with CONFIG.SWIDTH configured to 16 bit samples in a 16 bit half-frame on page 244 illustrates a configuration with identical sample and half-frame widths. The number of SCK pulses matches the number of sample bits. Aligned format, with CONFIG.SWIDTH configured to 16-bit samples in a 24-bit half-frame on page 244 illustrates a configuration with greater half-frame width than sample width. The number of SCK pulses are greater than the number of sample bits, with the sample being left-aligned in the half-frame.

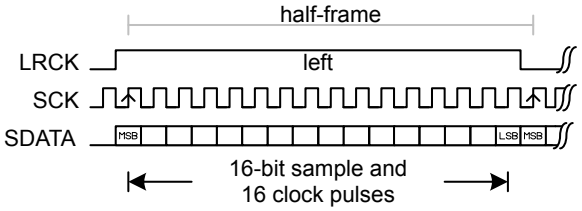

Figure 68: Aligned format, with CONFIG.SWIDTH configured to 16 bit samples in a 16 bit half-frame

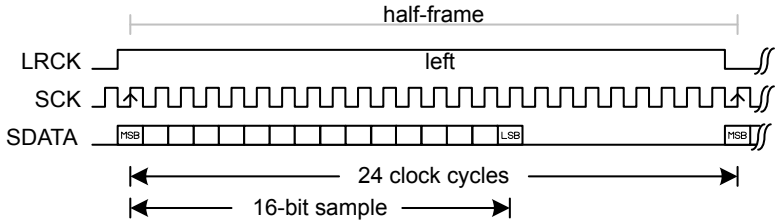

Figure 69: Aligned format, with CONFIG.SWIDTH configured to 16-bit samples in a 24-bit half-frame

The register CONFIG.FORMAT on page 260 is used to choose whether a word shall be aligned on the LRCK edge, or be delayed one bit period after this edge:

- When using Aligned format, the first bit in a half-frame gets sampled on the first rising edge of SCK following a LRCK edge, as illustrated in Aligned format. Identical sample width and half-frame width. Left sample on high level of LRCK on page 244. The left sample is transferred during the high half period of LRCK.
- When using I^{2}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, as illustrated in I^{2}S format. Identical sample width and half-frame width. Left sample on low level of LRCK on page 244. The left sample is transferred during the low half period of LRCK.

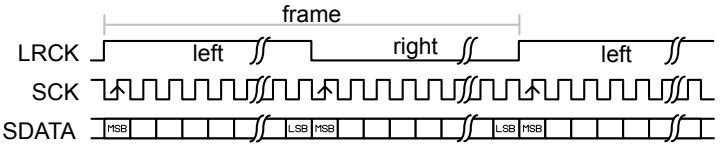

Figure 70: Aligned format. Identical sample width and half-frame width. Left sample on high level of LRCK

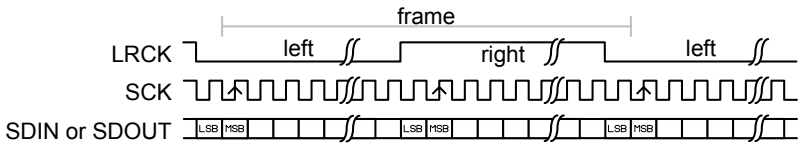

Figure 71: I^{2}S format. Identical sample width and half-frame width. Left sample on low level of LRCK

If the half-frame width differs from the sample width, the sample value can be either right or left-aligned inside a half-frame, as specified in CONFIG.ALIGN on page 260

- When using left-alignment, each half-frame starts with the MSB of the sample value, as illustrated by CONFIG.ALIGN set to left justified on page 244.
- When using right-alignment, each half-frame ends with the LSB of the sample value. This is illustrated in CONFIG.ALIGN set to right justified on page 245.

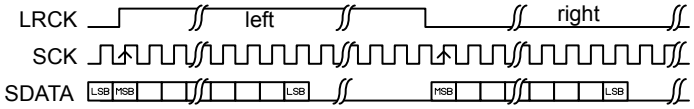

Figure 72: CONFIG.ALIGN set to left justified

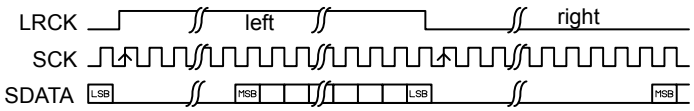

Figure 73: CONFIG.ALIGN set to right justified

Slave mode considerations

In Slave mode, the sample width does not need to equal the half-frame width, or even frame size. This means that there can be extra or fewer SCK pulses per half-frame than what the sample and half-frame widths specified in CONFIG.SWIDTH on page 259 require.

In cases where **left-alignment** is used, and the number of SCK pulses per half-frame is **higher** than the configured width, the following will apply:

- For data received on SDIN, all bits after the least significant bit (LSB) of the word value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the word value will be 0.

In cases where**left-alignment** is used, and the number of SCK pulses per frame is **lower** than the word width, the following will apply:

- Data sent and received on SDOUT and SDIN will be truncated with the LSBs being removed first.

In cases where **right-alignment** is used, and the number of SCK pulses per frame is **higher** than the configured width, the following will apply:

- For data received on SDIN, all bits before the MSB of the word value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the word value will be 0 (same behavior as for left-alignment).

In cases where **right-alignment** is used, and the number of SCK pulses per frame is **lower** than the configured width, the following will apply:

- Data received on SDIN will be sign-extended to the same number of bits as the sample width before being written to memory.
- Data sent on SDOUT will be truncated with the LSBs being removed first (same behavior as for left-alignment).

7.15.7 EasyDMA

The I^2S 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 261 and RXD.PTR on page 261. The memory pointed to by these pointers will only be read or written when TX or RX are enabled in CONFIG.TXEN on page 257, and CONFIG.RXEN on page 257.

The addresses written to the pointer registers TXD.PTR on page 261 and RXD.PTR on page 261 are double-buffered in hardware. These double buffers are updated for every number of transmitted data words given by RXTXD.MAXCNT on page 262 read from/written 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 261 is not pointing to the Data RAM region when transmission is enabled, or RXD.PTR on page 261 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 18 for more information about the different memory regions.

Due to the nature of I^2S, 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 262 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 one 32-bit sample, one right-aligned 24-bit sample sign extended to 32-bit, two 16-bit samples or four 8-bit samples.

In Stereo mode (CONFIG.CHANNELS on page 260=Stereo), the samples are stored as left and right sample pairs in memory. Memory mapping for 8-bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo. on page 246, Memory mapping for 16-bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo. on page 246 and Memory mapping for 24-bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo. on page 247 show how the samples are mapped to memory in this mode. The mapping is valid for both RX and TX.

In Mono mode (CONFIG.CHANNELS on page 260=Left or Right), RX sample from only one channel in the frame is stored in memory, the other channel sample is ignored. Memory mapping for 8-bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left. on page 246, Memory mapping for 16-bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left. on page 247 and Memory mapping for 24-bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left. on page 247 show how RX samples are mapped to memory in this mode. For TX, the same outgoing sample read from memory is transmitted on both left and right in a frame, resulting in a mono output stream.

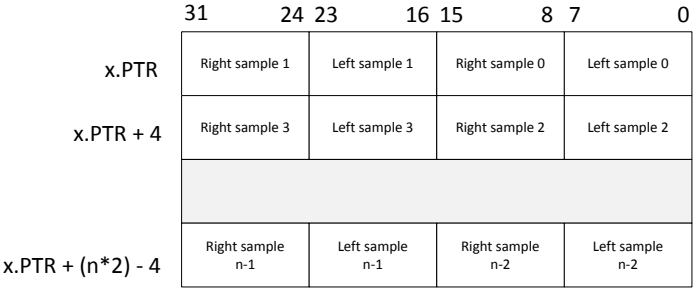

Figure 74: Memory mapping for 8-bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.

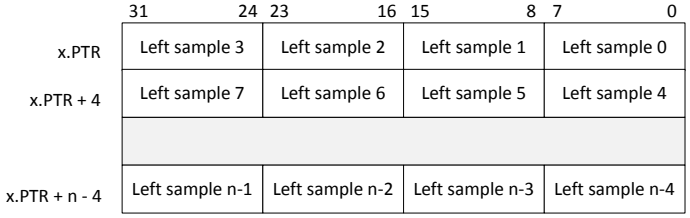

Figure 75: Memory mapping for 8-bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.

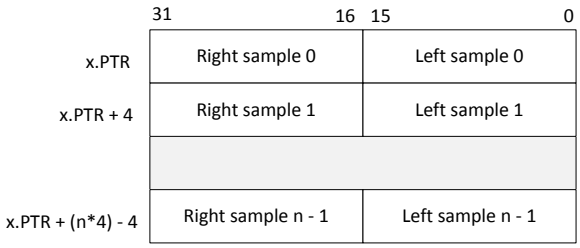

Figure 76: Memory mapping for 16-bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.

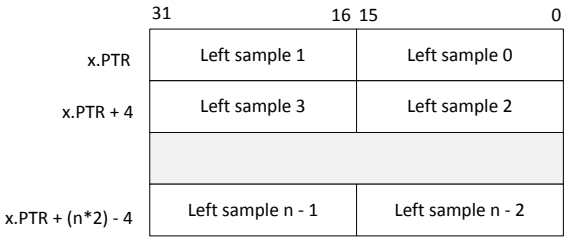

Figure 77: Memory mapping for 16-bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.

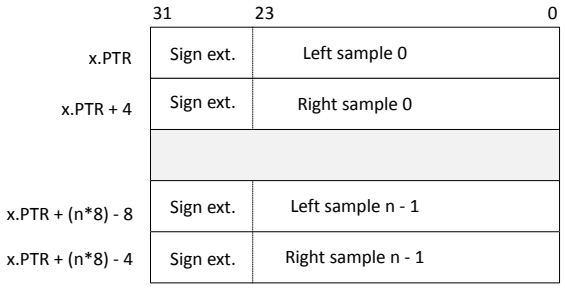

Figure 78: Memory mapping for 24-bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.

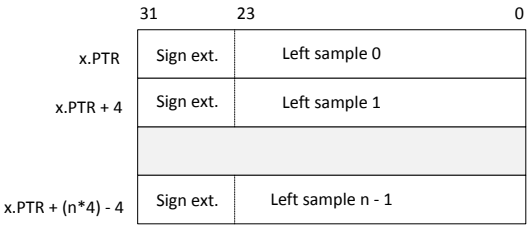

Figure 79: Memory mapping for 24-bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.

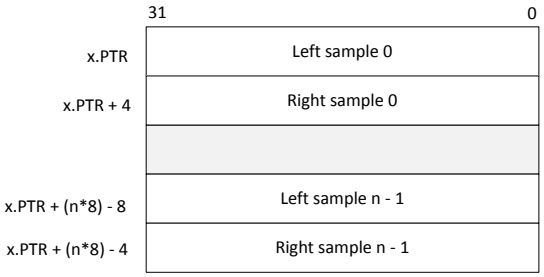

Figure 80: Memory mapping for 32-bit stereo. CONFIG.SWIDTH = 32Bit, CONFIG.CHANNELS = Stereo.

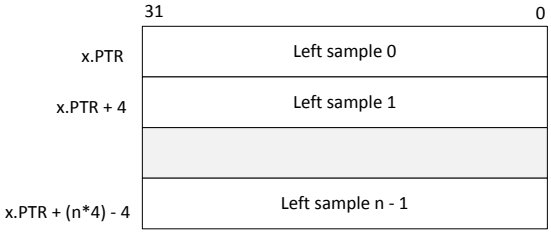

Figure 81: Memory mapping for 32-bit mono, left channel only. CONFIG.SWIDTH = 32Bit, CONFIG.CHANNELS = Left.

7.15.8 Module operation

Described here is a typical operating procedure for the I^{2}S module.

1. Configure the I^{2}S module using the CONFIG registers

```
// Enable reception
NRF_I2S->CONFIG.RXEN = (I2S_CONFIG_RXEN_RXEN_Enabled <<
                                        I2S_CONFIG_RXEN_RXEN_Pos);
// Enable transmission
NRF_I2S->CONFIG.TXEN = (I2S_CONFIG_TXEN_TXEN_Enabled <<
                                        I2S_CONFIG_TXEN_TXEN_Pos);
// Enable MCK generator
NRF_I2S->CONFIG.MCKEN = (I2S_CONFIG_MCKEN_MCKEN_Enabled <<
                                        I2S_CONFIG_MCKEN_MCKEN_Pos);
// MCKFREQ = 4 MHz
NRF_I2S->CONFIG.MCKFREQ = I2S_CONFIG_MCKFREQ_MCKFREQ_32MDIV8 <<
                                        I2S_CONFIG_MCKFREQ_MCKFREQ_Pos;
// Ratio = 256
NRF_I2S->CONFIG.RATIO = I2S_CONFIG_RATIO_RATIO_256X <<
                                        I2S_CONFIG_RATIO_RATIO_Pos;
// MCKFREQ = 4 MHz and Ratio = 256 gives sample rate = 15.625 ks/s
// Sample width = 16 bit
NRF_I2S->CONFIG.SWIDTH = I2S_CONFIG_SWIDTH_SWIDTH_16Bit <<
                                        I2S_CONFIG_SWIDTH_SWIDTH_Pos;
// Alignment = Left
NRF_I2S->CONFIG.ALIGN = I2S_CONFIG_ALIGN_ALIGN_Left <<
                                        I2S_CONFIG_ALIGN_ALIGN_Pos;
// Format = I2S
NRF_I2S->CONFIG.FORMAT = I2S_CONFIG_FORMAT_FORMAT_I2S <<
                                        I2S_CONFIG_FORMAT_FORMAT_Pos;
// Use stereo
NRF_I2S->CONFIG.CHANNELS = I2S_CONFIG_CHANNELS_CHANNELS_Stereo <<
                                        I2S_CONFIG_CHANNELS_CHANNELS_Pos;
```

2. Map IO pins using the PINSEL registers

```
// MCK routed to pin 0
NRF_I2S->PSEL.MCK = (0 << I2S_PSEL_MCK_PIN_Pos) |
                    (I2S_PSEL_MCK_CONNECT_Connected <<
                                                I2S_PSEL_MCK_CONNECT_Pos);
// SCK routed to pin 1
NRF_I2S->PSEL.SCK = (1 << I2S_PSEL_SCK_PIN_Pos) |
                    (I2S_PSEL_SCK_CONNECT_Connected <<
                                                I2S_PSEL_SCK_CONNECT_Pos);
// LRCK routed to pin 2
NRF_I2S->PSEL.LRCK = (2 << I2S_PSEL_LRCK_PIN_Pos) |
                     (I2S_PSEL_LRCK_CONNECT_Connected <<
                                                I2S_PSEL_LRCK_CONNECT_Pos);
// SDOUT routed to pin 3
NRF_I2S->PSEL.SDOUT = (3 << I2S_PSEL_SDOUT_PIN_Pos) |
                      (I2S_PSEL_SDOUT_CONNECT_Connected <<
                                                I2S_PSEL_SDOUT_CONNECT_Pos);
// SDIN routed on pin 4
NRF_I2S->PSEL.SDIN = (4 << I2S_PSEL_SDIN_PIN_Pos) |
                     (I2S_PSEL_SDIN_CONNECT_Connected <<
                                                I2S_PSEL_SDIN_CONNECT_Pos);
```

3. Configure TX and RX data pointers using the TXD, RXD and RXTXD registers

```
NRF_I2S->TXD.PTR = my_tx_buf;
NRF_I2S->RXD.PTR = my_rx_buf;
NRF_I2S->TXD.MAXCNT = MY_BUF_SIZE;
```

4. Enable the I^{2}S module using the ENABLE register

```
NRF_I2S->ENABLE = 1;
```

5. Start audio streaming using the START task

```
NRF_I2S->TASKS_START = 1;
```

6. Handle received and transmitted data when receiving the TXPTRUPD and RXPTRUPD events

```
if(NRF_I2S->EVENTS_TXPTRUPD  != 0)
{
    NRF_I2S->TXD.PTR = my_next_tx_buf;
    NRF_I2S->EVENTS_TXPTRUPD = 0;
}

if(NRF_I2S->EVENTS_RXPTRUPD != 0)
{
    NRF_I2S->RXD.PTR = my_next_rx_buf;
    NRF_I2S->EVENTS_RXPTRUPD = 0;
}
```

7.15.9 Pin configuration

The MCK, SCK, LRCK, SDIN and SDOUT signals associated with the I²S module are mapped to physical pins according to the pin numbers specified in the PSEL.x registers.

These pins are acquired whenever the I²S module is enabled through the register ENABLE on page 256.

When a pin is acquired by the I²S module, the direction of the pin (input or output) will be configured automatically, and any pin direction setting done in the GPIO module will be overridden. The directions for the various I²S pins are shown below in GPIO configuration before enabling peripheral (Master mode) on page 250 and GPIO configuration before enabling peripheral (Slave mode) on page 250.

To secure correct signal levels on the pins in System OFF mode, and when the I²S module is disabled, these pins must be configured in the GPIO peripheral directly.

I²S signal	I²S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Output	0	
SCK	As specified in PSEL.SCK	Output	0	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 85: GPIO configuration before enabling peripheral (Master mode)

I²S signal	I²S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 86: GPIO configuration before enabling peripheral (Slave mode)

7.15.10 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50028000 0x40028000	APPLICATION	I2S	I2S0 : S I2S0 : NS	US	SA	Inter-IC sound interface	

Table 87: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Starts continuous I2S transfer. Also starts MCK generator when this is enabled
TASKS_STOP	0x004		Stops I2S transfer and MCK generator. Triggering this task will cause the event STOPPED to be generated.
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
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 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.

Register	Offset	Security	Description
EVENTS_FRAMESTART	0x11C		Frame start event, generated on the active edge of LRCK
PUBLISH_RXPTRUPD	0x184		Publish configuration for event RXPTRUPD
PUBLISH_STOPPED	0x188		Publish configuration for event STOPPED
PUBLISH_TXPTRUPD	0x194		Publish configuration for event TXPTRUPD
PUBLISH_FRAMESTART	0x19C		Publish configuration for event FRAMESTART
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		Enable I2S module
CONFIG.MODE	0x504		I2S mode
CONFIG.RXEN	0x508		Reception (RX) enable
CONFIG.TXEN	0x50C		Transmission (TX) enable
CONFIG.MCKEN	0x510		Master clock generator enable
CONFIG.MCKFREQ	0x514		I2S clock generator control
CONFIG.RATIO	0x518		MCK / LRCK ratio
CONFIG.SWIDTH	0x51C		Sample width
CONFIG.ALIGN	0x520		Alignment of sample within a frame
CONFIG.FORMAT	0x524		Frame format
CONFIG.CHANNELS	0x528		Enable channels
CONFIG.CLKCONFIG	0x52C		Clock source selection for the I2S module
RXD.PTR	0x538		Receive buffer RAM start address.
TXD.PTR	0x540		Transmit buffer RAM start address
RXTXD.MAXCNT	0x550		Size of RXD and TXD buffers
PSEL.MCK	0x560		Pin select for MCK signal
PSEL.SCK	0x564		Pin select for SCK signal
PSEL.LRCK	0x568		Pin select for LRCK signal
PSEL.SDIN	0x56C		Pin select for SDIN signal
PSEL.SDOUT	0x570		Pin select for SDOUT signal

Table 88: Register overview

7.15.10.1 TASKS_START

Address offset: 0x000

Starts continuous I2S transfer. Also starts MCK generator when this is enabled

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Starts continuous I2S transfer. Also starts MCK generator when this is enabled
			Trigger	1	Trigger task

7.15.10.2 TASKS_STOP

Address offset: 0x004

Stops I2S transfer and MCK generator. Triggering this task will cause the event STOPPED to be generated.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stops I2S transfer and MCK generator. Triggering this task will cause the event STOPPED to be generated.
			Trigger	1	Trigger task

7.15.10.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.15.10.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.15.10.5 EVENTS_RXPTRUPD

Address offset: 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 received on the SDIN 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXPTRUPD			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 received on the SDIN pin.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.15.10.6 EVENTS_STOPPED

Address offset: 0x108

I2S transfer stopped.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			I2S transfer stopped.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.15.10.7 EVENTS_TXPTRUPD

Address offset: 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.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXPTRUPD			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.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.15.10.8 EVENTS_FRAMESTART

Address offset: 0x11C

Frame start event, generated on the active edge of LRCK

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_FRAMESTART			Frame start event, generated on the active edge of LRCK
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.15.10.9 PUBLISH_RXPTRUPD

Address offset: 0x184

Publish configuration for event RXPTRUPD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXPTRUPD will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.15.10.10 PUBLISH_STOPPED

Address offset: 0x188

Publish configuration for event STOPPED

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.15.10.11 PUBLISH_TXPTRUPD

Address offset: 0x194

Publish configuration for event TXPTRUPD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXPTRUPD will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.15.10.12 PUBLISH_FRAMESTART

Address offset: 0x19C

Publish configuration for event FRAMESTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event FRAMESTART will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.15.10.13 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 F C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Enable or disable interrupt for event RXPTRUPD
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXPTRUPD			Enable or disable interrupt for event TXPTRUPD
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	FRAMESTART			Enable or disable interrupt for event FRAMESTART
			Disabled	0	Disable
			Enabled	1	Enable

7.15.10.14 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 F C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Write '1' to enable interrupt for event RXPTRUPD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled

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

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
F	RW	TXPTRUPD			Write '1' to enable interrupt for event TXPTRUPD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	FRAMESTART			Write '1' to enable interrupt for event FRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.15.10.15 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 F C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
B	RW	RXPTRUPD			Write '1' to disable interrupt for event RXPTRUPD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXPTRUPD			Write '1' to disable interrupt for event TXPTRUPD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	FRAMESTART			Write '1' to disable interrupt for event FRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.15.10.16 ENABLE

Address offset: 0x500

Enable I2S module

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable I2S module
			Disabled	0	Disable
			Enabled	1	Enable

7.15.10.17 CONFIG.MODE

Address offset: 0x504

I2S mode

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

ID	R/W	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

7.15.10.18 CONFIG.RXEN

Address offset: 0x508

Reception (RX) enable

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

ID	R/W	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.

7.15.10.19 CONFIG.TXEN

Address offset: 0x50C

Transmission (TX) enable

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

ID	R/W	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.

7.15.10.20 CONFIG.MCKEN

Address offset: 0x510

Master clock generator enable

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

7.15.10.21 CONFIG.MCKFREQ

Address offset: 0x514

I2S clock generator control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x20000000				**0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MCKFREQ			I2S MCK frequency configuration NOTE: Enumerations are deprecated, use MCKFREQ equation. NOTE: The 12 least significant bits of the register are ignored and shall be set to zero.
			32MDIV2	0x80000000	32 MHz / 2 = 16.0 MHz Deprecated, use MCKFREQ equation.
			32MDIV3	0x50000000	32 MHz / 3 = 10.6666667 MHz Deprecated, use MCKFREQ equation.
			32MDIV4	0x40000000	32 MHz / 4 = 8.0 MHz Deprecated, use MCKFREQ equation.
			32MDIV5	0x30000000	32 MHz / 5 = 6.4 MHz Deprecated, use MCKFREQ equation.
			32MDIV6	0x28000000	32 MHz / 6 = 5.3333333 MHz Deprecated, use MCKFREQ equation.
			32MDIV8	0x20000000	32 MHz / 8 = 4.0 MHz Deprecated, use MCKFREQ equation.
			32MDIV10	0x18000000	32 MHz / 10 = 3.2 MHz Deprecated, use MCKFREQ equation.
			32MDIV11	0x16000000	32 MHz / 11 = 2.9090909 MHz Deprecated, use MCKFREQ equation.
			32MDIV15	0x11000000	32 MHz / 15 = 2.1333333 MHz Deprecated, use MCKFREQ equation.
			32MDIV16	0x10000000	32 MHz / 16 = 2.0 MHz Deprecated, use MCKFREQ equation.
			32MDIV21	0x0C000000	32 MHz / 21 = 1.5238095 MHz Deprecated, use MCKFREQ equation.
			32MDIV23	0x0B000000	32 MHz / 23 = 1.3913043 MHz Deprecated, use MCKFREQ equation.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x20000000				**0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			32MDIV30	0x08800000	32 MHz / 30 = 1.0666667 MHz Deprecated, use MCKFREQ equation.
			32MDIV31	0x08400000	32 MHz / 31 = 1.0322581 MHz Deprecated, use MCKFREQ equation.
			32MDIV32	0x08000000	32 MHz / 32 = 1.0 MHz Deprecated, use MCKFREQ equation.
			32MDIV42	0x06000000	32 MHz / 42 = 0.7619048 MHz Deprecated, use MCKFREQ equation.
			32MDIV63	0x04100000	32 MHz / 63 = 0.5079365 MHz Deprecated, use MCKFREQ equation.
			32MDIV125	0x020C0000	32 MHz / 125 = 0.256 MHz Deprecated, use MCKFREQ equation.

7.15.10.22 CONFIG.RATIO

Address offset: 0x518

MCK / LRCK ratio

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000006				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RATIO			MCK / LRCK ratio
			32X	0	LRCK = MCK / 32
			48X	1	LRCK = MCK / 48
			64X	2	LRCK = MCK / 64
			96X	3	LRCK = MCK / 96
			128X	4	LRCK = MCK / 128
			192X	5	LRCK = MCK / 192
			256X	6	LRCK = MCK / 256
			384X	7	LRCK = MCK / 384
			512X	8	LRCK = MCK / 512

7.15.10.23 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 A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	SWIDTH			Sample and half-frame width
			8Bit	0	8 bit sample.
			16Bit	1	16 bit sample.
			24Bit	2	24 bit sample.
			32Bit	3	32 bit sample.
			8BitIn16	4	8 bit sample in a 16-bit half-frame.
			8BitIn32	5	8 bit sample in a 32-bit half-frame.
			16BitIn32	6	16 bit sample in a 32-bit half-frame.
			24BitIn32	7	24 bit sample in a 32-bit half-frame.

7.15.10.24 CONFIG.ALIGN

Address offset: 0x520

Alignment of sample within a frame

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ALIGN			Alignment of sample within a frame
			Left	0	Left-aligned.
			Right	1	Right-aligned.

7.15.10.25 CONFIG.FORMAT

Address offset: 0x524

Frame format

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

ID	R/W	Field	Value ID	Value	Description
A	RW	FORMAT			Frame format
			I2S	0	Original I2S format.
			Aligned	1	Alternate (left- or right-aligned) format.

7.15.10.26 CONFIG.CHANNELS

Address offset: 0x528

Enable channels

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHANNELS			Enable channels
			Stereo	0	Stereo.
			Left	1	Left only.
			Right	2	Right only.

7.15.10.27 CONFIG.CLKCONFIG

Address offset: 0x52C

Clock source selection for the 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CLKSRC			Clock source selection
			PCLK32M	0	32MHz peripheral clock
			ACLK	1	Audio PLL clock
B	RW	BYPASS			Bypass clock generator. MCK will be equal to source input. If bypass is enabled the MCKFREQ setting has no effect.
			Disable	0	Disable bypass
			Enable	1	Enable bypass

7.15.10.28 RXD.PTR

Address offset: 0x538

Receive buffer RAM start address.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Receive buffer Data RAM start address. When receiving, words containing samples will be written to this address. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

7.15.10.29 TXD.PTR

Address offset: 0x540

Transmit buffer RAM start address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	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. **Note:** See the memory chapter for details about which memories are available for EasyDMA.

7.15.10.30 RXTXD.MAXCNT

Address offset: 0x550

Size of RXD and TXD buffers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT			Size of RXD and TXD buffers in number of 32 bit words

7.15.10.31 PSEL.MCK

Address offset: 0x560

Pin select for MCK signal

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

7.15.10.32 PSEL.SCK

Address offset: 0x564

Pin select for SCK signal

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

7.15.10.33 PSEL.LRCK

Address offset: 0x568

Pin select for LRCK signal

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

7.15.10.34 PSEL.SDIN

Address offset: 0x56C

Pin select for SDIN signal

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

7.15.10.35 PSEL.SDOUT

Address offset: 0x570

Pin select for SDOUT signal

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

7.15.11 Electrical specification

7.15.11.1 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	50			ns
t_{H_SDOUT}	SDOUT hold time after SCK falling	13			ns
t_{SCK_LRCK}	SCLK falling to LRCK edge	-5	0	5	ns
f_{MCK}	MCK frequency			12288	kHz
f_{LRCK}	LRCK frequency			96	kHz
f_{SCK}	SCK frequency			8000	kHz
DC_{CK}	Clock duty cycle (MCK, LRCK, SCK)	45		55	%

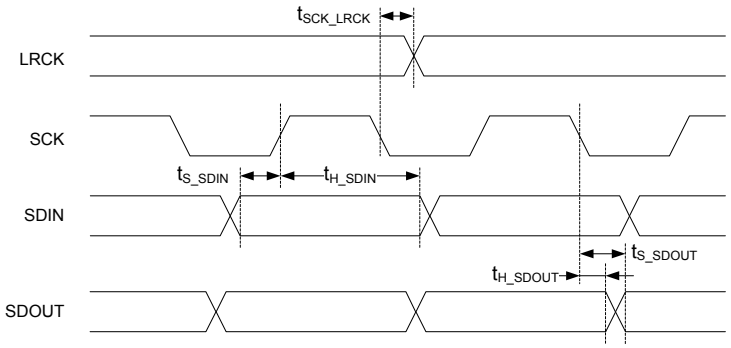

Figure 82: I2S timing diagram

7.16 IPC — Interprocessor communication

The interprocessor communication (IPC) peripheral is used to send and receive events between MCUs in the system.

The following figure illustrates the IPC peripheral in a multi-MCU system, where each MCU has one dedicated IPC peripheral. The IPC peripheral can be used to send and receive events to and from other IPC peripherals.

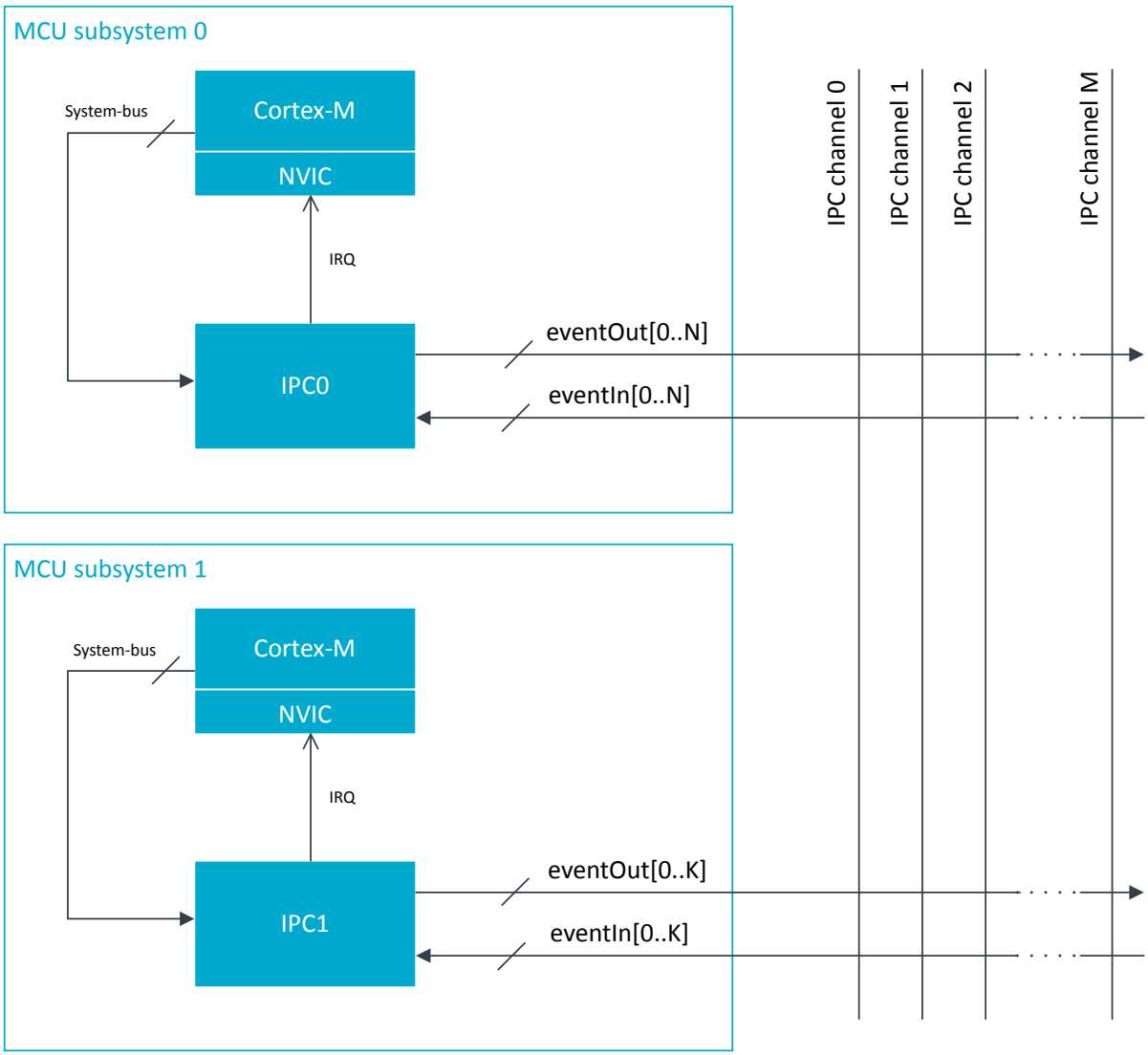

Figure 83: IPC block diagram

An instance of the IPC peripheral can have multiple SEND tasks and RECEIVE events. A single SEND task can be configured to signal an event on one or more IPC channels, and a RECEIVE event can be configured to listen on one or more IPC channels. The IPC channels that are triggered in a SEND task can be configured through the SEND_CNF registers, and the IPC channels that trigger a RECEIVE event are configured through the RECEIVE_CNF registers. The figure below illustrates how the SEND_CNF and RECEIVE_CNF registers work. Both the SEND task and the RECEIVE event can be connected to all IPC channels.

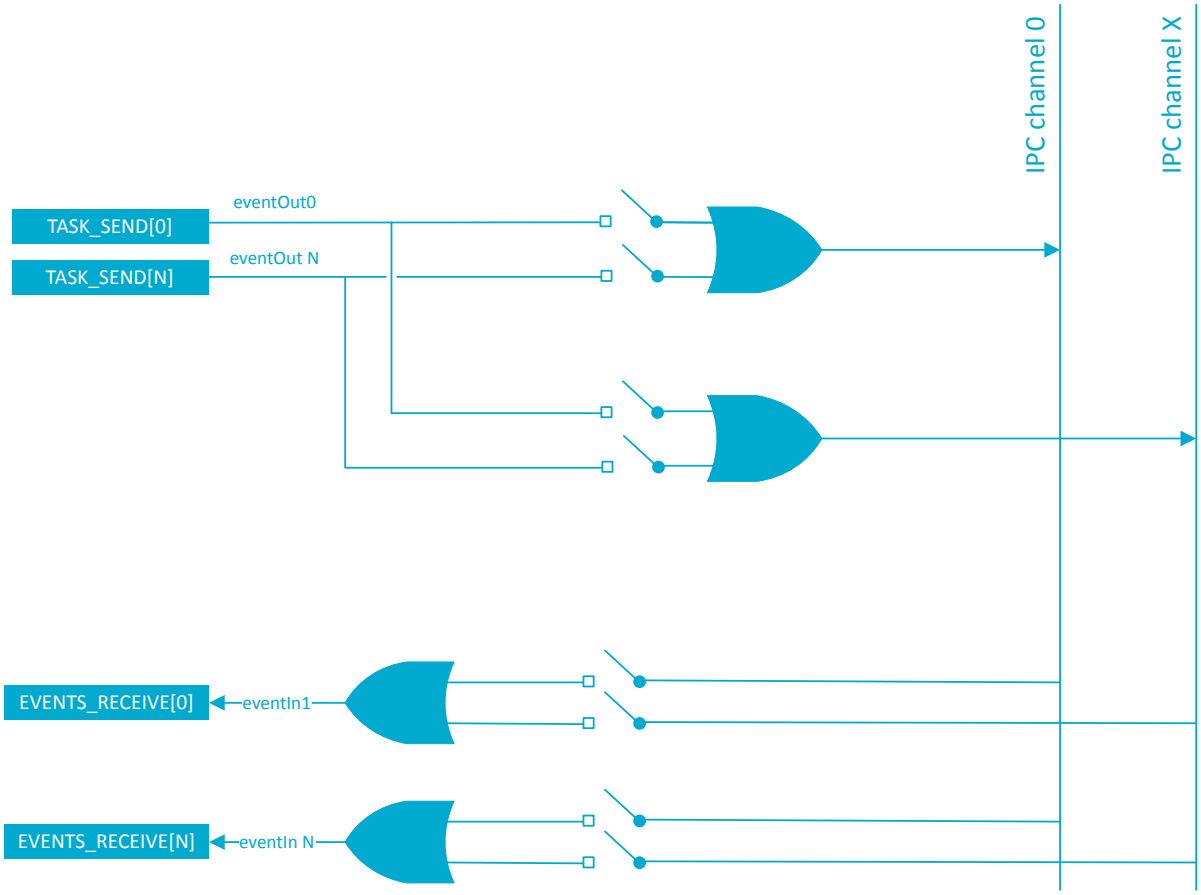

Figure 84: IPC registers SEND_CNF and RECEIVE_CNF

A SEND task can be viewed as broadcasting events onto one or more IPC channels, and a RECEIVE event can be seen as subscribing to a subset of IPC channels. It is possible for multiple IPCs to trigger events onto the same PPI channel at the same time. When two or more events on the same channel occur within t_{IPC}, the events may be merged into a single event seen from the IPC receiver. One of the events can therefore be lost. To prevent this, the user must ensure that events on the same IPC channel do not occur within t_{IPC} of each other. When implementing firmware data structures, such as queues or mailboxes, this can be done by using one IPC channel for acknowledgements.

An IPC event often does not contain any data itself, it is used to signal other MCUs that something has occurred. Data can be shared through shared memory, for example in the form of a software implemented mailbox, or command/event queues. It is up to software to assign a logical functionality to an IPC channel. For instance, one IPC channel can be used to signal that a command is ready to be executed, and any processor in the system can subscribe to that particular IPC channel and decode/execute the command.

General purpose memory

The GPMEM registers can be used freely to store information. These registers are accessed like any other of the IPC peripheral's registers. Note that the contents of the GPMEM registers are not shared between the instances of the peripherals. I.e. writing the GPMEM register of one peripheral does not change the value in another.

7.16.1 IPC and PPI connections

The IPC SEND tasks and RECEIVE events can be connected through PPI channels. This makes it possible to relay events from peripherals in one MCU to another, without CPU involvement.

Figure below illustrates a timer COMPARE event that is relayed from one MCU to IPC using PPI, then back into a timer CAPTURE event in another MCU.

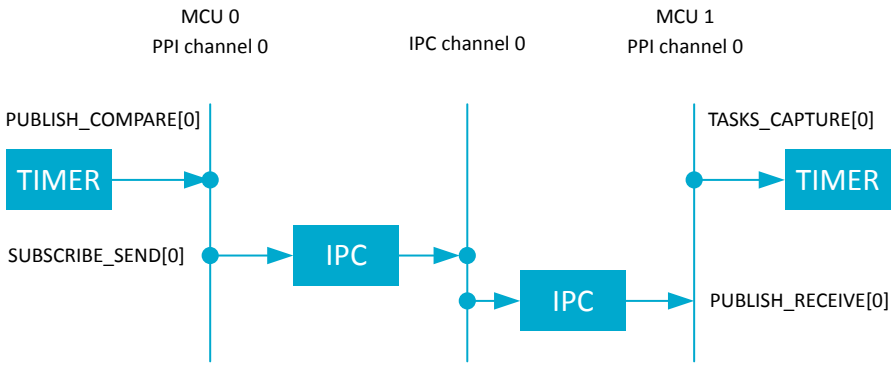

Figure 85: Example of PPI and IPC connections

7.16.2 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5002A000 0x4002A000	APPLICATION	IPC	IPC : S IPC : NS	US	NA	Interprocessor communication	
0x41012000	NETWORK	IPC	IPC	NS	NA	Interprocessor communication	

Table 89: Instances

Register	Offset	Security	Description
TASKS_SEND[n]	0x000		Trigger events on IPC channel enabled in SEND_CNF[n]
SUBSCRIBE_SEND[n]	0x080		Subscribe configuration for task SEND[n]
EVENTS_RECEIVE[n]	0x100		Event received on one or more of the enabled IPC channels in RECEIVE_CNF[n]
PUBLISH_RECEIVE[n]	0x180		Publish configuration for event RECEIVE[n]
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
INTPEND	0x30C		Pending interrupts
SEND_CNF[n]	0x510		Send event configuration for TASKS_SEND[n]
RECEIVE_CNF[n]	0x590		Receive event configuration for EVENTS_RECEIVE[n]
GPMEM[n]	0x610		General purpose memory

Table 90: Register overview

7.16.2.1 TASKS_SEND[n] (n=0..15)

Address offset: 0x000 + (n × 0x4)

Trigger events on IPC channel enabled in SEND_CNF[n]

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	W	TASKS_SEND			Trigger events on IPC channel enabled in SEND_CNF[n]
			Trigger	1	Trigger task

7.16.2.2 SUBSCRIBE_SEND[n] (n=0..15)

Address offset: 0x080 + (n × 0x4)

Subscribe configuration for task SEND[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SEND[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.16.2.3 EVENTS_RECEIVE[n] (n=0..15)

Address offset: 0x100 + (n × 0x4)

Event received on one or more of the enabled IPC channels in RECEIVE_CNF[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RECEIVE			Event received on one or more of the enabled IPC channels in RECEIVE_CNF[n]
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.16.2.4 PUBLISH_RECEIVE[n] (n=0..15)

Address offset: 0x180 + (n × 0x4)

Publish configuration for event RECEIVE[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RECEIVE[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.16.2.5 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-P	RW	RECEIVE[i] (i=0..15)			Enable or disable interrupt for event RECEIVE[i]
			Disabled	0	Disable
			Enabled	1	Enable

7.16.2.6 INTENSET

Address offset: 0x304

Enable interrupt

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

7.16.2.7 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.16.2.8 INTPEND

Address offset: 0x30C

Pending interrupts

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-P	R	RECEIVE[i] (i=0..15)			Read pending status of interrupt for event RECEIVE[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

7.16.2.9 SEND_CNF[n] (n=0..15)

Address offset: 0x510 + (n × 0x4)

Send event configuration for TASKS_SEND[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-P	RW	CHEN[i] (i=0..15)			Enable broadcasting on IPC channel i
			Disable	0	Disable broadcast
			Enable	1	Enable broadcast

7.16.2.10 RECEIVE_CNF[n] (n=0..15)

Address offset: 0x590 + (n × 0x4)

Receive event configuration for EVENTS_RECEIVE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-P	RW	CHEN[i] (i=0..15)			Enable subscription to IPC channel i
			Disable	0	Disable events
			Enable	1	Enable events

7.16.2.11 GPMEM[n] (n=0..1)

Address offset: 0x610 + (n × 0x4)

General purpose memory

Retained only 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				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	GPMEM			General purpose memory

7.16.3 Electrical specification

7.16.3.1 IPC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{IPC}	Time window during which IPC events can be merged	..	..	..	µs

7.17 KMU — Key management unit

The key management unit (KMU) enforces access policies to a subset region of user information configuration register (UICR). This subset region is used for storing cryptographic key values inside the key slots, which the CPU has no access to.

In total there are 128 key slots available, where each key slot can store one 128-bit key value together with an access policy and a destination address for the key value. Multiple key slots can be combined in order

to support key sizes larger than 128 bits. The access policy of a key slot governs if and how a key value can be used, while the destination address determines where in the memory map the KMU pushes the key value upon a request from the CPU.

Key slots can be configured to be pushed directly into write-only key registers in cryptographic accelerators, like e.g. CryptoCell, without exposing the key value itself to the CPU. This enables the CPU to use the key values stored inside the key slots for cryptographic operations without being exposed to the key value.

Access to the KMU, and the key slots in the UICR, is only allowed from secure mode.

7.17.1 Functional view

From a functional view the UICR is divided into two different regions, one-time programmable (OTP) memory and key storage.

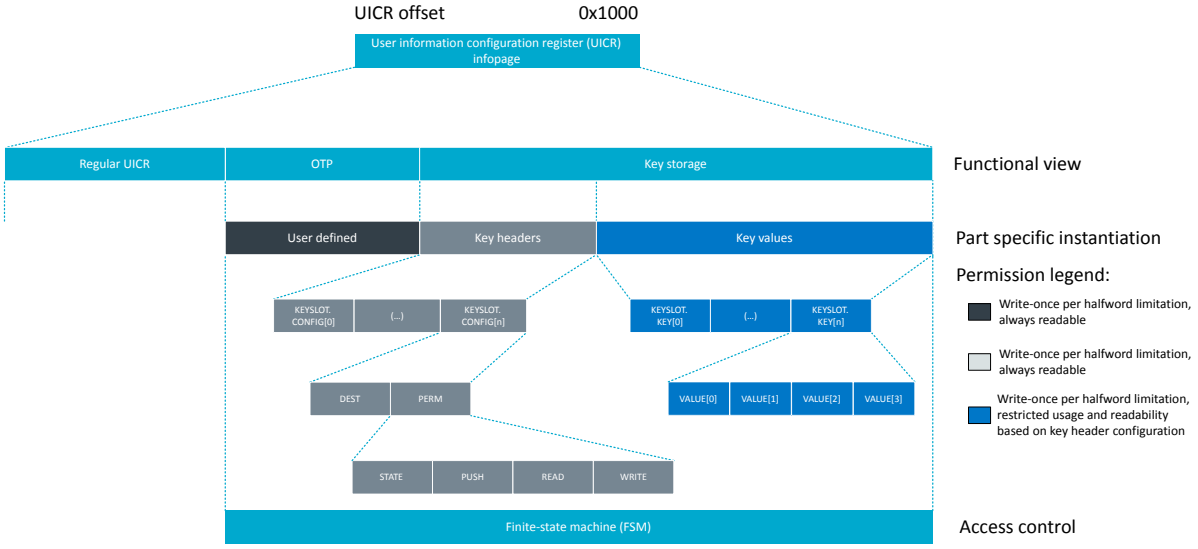

Figure 86: Memory map overview

One-time programmable (OTP) memory is typically used for holding values that are written once, and then never to be changed again throughout the product lifetime. The OTP region of UICR is emulated by placing a write-once per halfword limitation on registers defined here.

The key storage region contains multiple key slots, where each slot consists of a key header and an associated key value. The key value is limited to 128 bits. Any key size greater than 128 bits must be divided and distributed over multiple key slot instances.

Key headers are allocated an address range of 0x400 in the UICR memory map, allowing a total of 128 keys to be addressable inside the key storage region.

Note: The use of the key storage region in UICR should be limited to keys with a certain life span, and not per-session derived keys where the CPU is involved in the key exchange.

7.17.2 Access control

Access control to the underlying UICR infopage in flash is enforced by a hardware finite-state machine (FSM). The FSM can allow or block transactions, depending both on the security of the transaction (secure or non-secure) and on the type of register being written and/or read.

Access type	Key headers	Key values
Read	Allowed	Restricted
Write	Restricted	Restricted

Table 91: Access control

Any restricted access requires an explicit key slot selection through the KMU register interface. Any illegal access to restricted key slot registers will be blocked and word `0xDEADDEAD` will be returned on the AHB.

The OTP region has individual access control behavior, while access control to the key storage region is configured on a per key slot basis. The KMU FSM operates on only one key slot instance at a time, and the permissions and the usage restriction for a key value associated with a key slot can be configured individually.

Note: Even if the KMU can be configured as non-secure, all non-secure transactions will be blocked.

7.17.3 Protecting the UICR content

The UICR content can be protected against device-internal NVMC.ERASEALL requests, in addition to device-external ERASEALL requests, through the CTRL-AP interface. This feature is useful if the firmware designers want to prevent the OTP region from being erased.

Since enabling this step will permanently disable erase for the UICR, the procedure requires an implementation defined 32-bit word to be written into the UICR's ERASEPROTECT register.

In case of a field return handling, it is still possible to erase the UICR even if the ERASEPROTECT is set. If this functionality is desired, the secure boot code must implement a secure communication channel over the CTRL-AP mailbox interface. Upon successful authentication of the external party, the secure boot code can temporarily re-enable the CTRL-AP ERASEALL functionality.

7.17.4 Usage

This section describes the specific KMU and UICR behavior in more detail, to help the reader get a better overview of KMU's features and the intended usage.

7.17.4.1 OTP

The OTP region of the UICR contains a user-defined static configuration of the device. The KMU emulates the OTP functionality by placing a write-once per halfword limitation of registers defined in this region, i.e. only halfwords containing all '1's can be written.

An OTP write transaction must consist of a full 32-bit word. Both halfwords can either be written simultaneously or one at a time. The KMU FSM will block any write to a halfword in the OTP region, if the initial value of this halfword is not `0xFFFF`. When writing halfwords one at a time, the non-active halfword must be masked as `0xFFFF`, otherwise the request will be blocked. For example, writing `0x1234XXXX` to an OTP destination address which already contains the value `0xFFFFAABB`, must be configured as `0x1234FFFF`. The OTP destination address will contain the value `0x1234AABB` after both write transactions have been processed.

The KMU will also only allow secure AHB write transactions into the OTP region of the UICR. Any AHB write transaction to this region that does not satisfy the above requirements will be ignored, and the STATUS.BLOCKED register will be set to '1'.

7.17.4.2 Key storage

The key storage region of the UICR can contain multiple keys of different type, including symmetrical keys, hashes, public/private key pairs and other device secrets. One of the key features of the KMU, is that these

device secrets can be installed and made available for use in cryptographic operations without revealing the actual secret values.

Keys in this region will typically have a certain life span. The region is not designed to be used for per-session derived keys where the non-secure side (i.e. application) is participating in the key exchange.

All key storage is done through the concept of multiple key slots, where each key slot instance consists of one key header and an associated key value. Each key header supports the configuration of usage permissions and an optional secure destination address.

The key header secure destination address option enables the KMU to push the associated key value over a dedicated secure APB to a pre-configured secure location within the memory map. Such locations typically include a write-only key register of the hardware cryptograhic accelerator, allowing the KMU to distribute keys within the system without compromising the key values.

One key slot instance can store a key value of maximum 128 bits. If a key size exceeds this limit, the key value itself must be split over multiple key slot instances.

The following usage and read permissions scheme is applicable for each key slot:

State	Push	Read	Write	Description
Active (1)	Enabled (1)	Enabled (1)	Enabled (1)	Default flash erase value. Key slot cannot be pushed, write is enabled.
Active (1)	Enabled (1)	Enabled (1)	Disabled (0)	Key slot is active, push is enabled. Key slot VALUE registers can be read, but write is disabled.
Active (1)	Enabled (1)	Disabled (0)	Disabled (0)	Key slot is active, push is enabled. Read and write to key slot VALUE registers are disabled.
Active (1)	Disabled (0)	Enabled (1)	Disabled (0)	Key slot is active, push is disabled. Key slot VALUE registers can be read, but write is disabled.
Revoked (0)	-	-	-	Key slot is revoked and key value is set to zero. Cannot be read or pushed over secure APB regardless of the permission settings.

Table 92: Valid key slot permission schemes

7.17.4.2.1 Selecting a key slot

The KMU FSM is designed to process only one key slot at a time, effectively operating as a memory protection unit for the key storage region. Whenever a key slot is selected, the KMU will allow access to writing, reading, and/or pushing the associated key value according to the selected slot configuration.

A key slot must be selected prior to use, by writing the key slot ID into the KMU SELECTKEYSLOT register. Because the reset value of this register is `0x00000000`, there is no key slot associated with ID=0 and no slot is selected by default. All key slots are addressed using IDs from 1 to 128.

SELECTED status is set when a key slot is selected, and a read or write acccess to that keyslot occurs.

BLOCKED status is set when any illegal access to key slot registers is detected.

When the use of the particular key slot is stopped, the key slot selection in SELECTKEYSLOT must be set back to `0`.

By default, all KMU key slots will consist of a 128-bit key value of '1's, where the key headers have no secure destination address, or any usage and read restrictions.

7.17.4.2.2 Writing to a key slot

Writing a key slot into UICR is a five-step process.

1. Select which key slot the KMU shall operate on by writing the desired key slot ID into the KMU SELECTKEYSLOT register. The selected key slot must be empty in order to add a new entry to UICR.
2. If the key value shall be pushable over secure APB, the destination address of the recipient must be configured in register KEYSLOT.CONFIG[ID-1].DEST.

3. Write the 128-bit key value into KEYSLOT.KEY[ID-1].VALUE[0-3].
4. Write the desired key slot permissions into KEYSLOT.CONFIG[ID-1].PERM, including any applicable usage restrictions.
5. Select key slot 0.

In case the total key size is greater than 128 bits, the key value itself must be split into 128-bit segments and written to multiple key slot instances. Steps 1 through 5 above must be repeated for the entire key size.

Note: If a key slot is configured as readable, and KEYSLOT.CONFIG[ID-1].DEST is not to be used, it is recommended to disable the push bit in KEYSLOT.CONFIG[ID-1].PERM when configuring key slot permissions.

Note: A key value distributed over multiple key slots should use the same key slot configuration in its key headers, but the secure destination address for each key slot instance must be incremented by 4 words (128 bits) for each key slot instance spanned.

Note: Write to flash must be enabled in NVMC->CONFIG prior to writing keys to flash, and subsequently disabled once writing is complete.

Steps 1 through 5 above will be blocked if any of the following violations are detected:

- No key slot selected
- Non-empty key slot selected
- NVM destination address not empty
- AHB write to KEYSLOT.KEY[ID-1].VALUE[0-3] registers not belonging to selected key slot

7.17.4.2.3 Reading a key value

Key slots that are configured as readable can have their key value read directly from the UICR memory map by the CPU.

Readable keys are typically used during the secure boot sequence, where the CPU is involved in falsifying or verifying the integrity of the system. Since the CPU is involved in this decision process, it makes little sense not to trust the CPU having access to the actual key value but ultimately trust the decision of the integrity check. Another use-case for readable keys is if the key type in question does not have a HW peripheral in the platform that is able to accept such keys over secure APB.

Reading a key value from the UICR is a three-step process:

1. Select the key slot which the KMU shall operate on by writing the desired key slot ID into KMU->SELECTKEYSLOT.
2. If STATE and READ permission requirements are fulfilled as defined in KEYSLOT.CONFIG[ID-1].PERM, the key value can be read from region KEYSLOT.KEY[ID-1].VALUE[0-3] for selected key slot.
3. Select key slot 0.

Step 2 will be blocked and word 0xDEADDEAD will be returned on AHB if any of the following violations are detected:

- No key slot selected
- Key slot not configured as readable
- Key slot is revoked
- AHB read to KEYSLOT.KEY[ID-1].VALUE[0-3] registers not belonging to selected key slot

7.17.4.2.4 Push over secure APB

Key slots that are configured as non-readable cannot be read by the CPU regardless of the mode the system is in, and must be pushed over secure APB in order to use the key value for cryptographic operations.

The secure APB destination address is set in the key slot configuration DEST register. Such destination addresses are typically write-only key registers in a hardware cryptographic accelerators memory map. The secure APB allows key slots to be utilized by the software side, without exposing the key value itself.

Pushing a key slot over secure APB is a four-step process:

1. Select the key slot on which the KMU shall operate by writing the desired key slot ID into KMU->SELECTKEYSLOT.
2. Start TASKS_PUSH_KEYSLOT to initiate a secure APB transaction, writing the 128-bit key value associated with the selected key slot into address defined in KEYSLOT.CONFIG[ID-1].DEST.
3. After completing the secure APB transaction, the 128-bit key value is ready for use by the peripheral and EVENTS_KEYSLOT_PUSHED is triggered.
4. Select key slot 0.

> **Note:** If a key value is distributed over multiple key slots due to its key size, exceeding the maximum 128-bit key value limitation, then each distributed key slot must be pushed individually in order to transfer the entire key value over secure APB.

Step 3 will trigger other events than EVENTS_KEYSLOT_PUSHED if the following violations are detected:

- EVENTS_KEYSLOT_ERROR:
  - If no key slot is selected
  - If a key slot has no destination address configured
  - If when pushing a key slot, flash or peripheral returns an error
  - If pushing a key slot when push permissions are disabled
  - If attempting to push a key slot with default permissions
- EVENTS_KEYSLOT_REVOKED if a key slot is marked as revoked in its key header configuration

7.17.4.2.5 Revoking the key slots

All key slots within the key storage area can be marked as revoked.

To revoke any key slots, write to the STATE field in the KEYSLOT.CONFIG[ID-1].PERM register. The following rules apply to keys that have been revoked:

1. Key slots that have the PUSH field enabled in PERM register can no longer be pushed. If a revoked key slot is selected and task TASKS_PUSH_KEYSLOT is started, the event EVENTS_KEYSLOT_REVOKED is triggered.
2. Key slots that have the READ field enabled in PERM register are invalidated. Any read operation to a revoked key value will return the fixed value `3735936685`.
3. Previously pushed key values stored in a peripheral write-only key register are not affected by key revocation. If secure code wants to enforce that a revoked key is no longer usable by a peripheral for cryptographic operations, the secure code should disable or reset the peripheral in question.

> **Note:** If a key slot is revoked, the KMU will automatically zeroize the associated VALUE registers.

7.17.4.3 STATUS register

The KMU uses a STATUS register to indicate its status of operation. The SELECTED bit will be asserted whenever the currently selected key slot is successfully read from or written to.

All read or write operations to other key slots than what is currently selected in KMU->SELECTKEYSLOT will assert the BLOCKED bit. The BLOCKED bit will also be asserted if the KMU fails to select a key slot, or

if a request has been blocked due to an access violation. Normal operation using the KMU should never trigger the BLOCKED bit. If this bit is triggered during the development phase, it indicates that the code is using the KMU incorrectly.

The STATUS register is reset every time register SELECTKEYSLOT is written.

7.17.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50039000 0x40039000	APPLICATION	KMU	KMU : S KMU : NS	SPLIT	NA	Key management unit	

Table 93: Instances

Register	Offset	Security	Description
TASKS_PUSH_KEYSLOT	0x0000		Push a key slot over secure APB
EVENTS_KEYSLOT_PUSHED	0x100		Key slot successfully pushed over secure APB
EVENTS_KEYSLOT_REVOKED	0x104		Key slot has been revoked and cannot be tasked for selection
EVENTS_KEYSLOT_ERROR	0x108		No key slot selected, no destination address defined, or error during push operation
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
INTPEND	0x30C		Pending interrupts
STATUS	0x40C		Status bits for KMU operation
SELECTKEYSLOT	0x500		Select key slot to be read over AHB or pushed over secure APB when TASKS_PUSH_KEYSLOT is started

Table 94: Register overview

7.17.5.1 TASKS_PUSH_KEYSLOT

Address offset: 0x0000

Push a key slot over secure APB

Bit number	31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value	Description
A	W	TASKS_PUSH_KEYSLOT			Push a key slot over secure APB
			Trigger	1	Trigger task

7.17.5.2 EVENTS_KEYSLOT_PUSHED

Address offset: 0x100

Key slot successfully pushed over secure APB

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_KEYSLOT_PUSHED			Key slot successfully pushed over secure APB
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.17.5.3 EVENTS_KEYSLOT_REVOKED

Address offset: 0x104

Key slot has been revoked and cannot be tasked for 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_KEYSLOT_REVOKED			Key slot has been revoked and cannot be tasked for selection
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.17.5.4 EVENTS_KEYSLOT_ERROR

Address offset: 0x108

No key slot selected, no destination address defined, or error during push operation

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_KEYSLOT_ERROR			No key slot selected, no destination address defined, or error during push operation
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.17.5.5 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	KEYSLOT_PUSHED			Enable or disable interrupt for event KEYSLOT_PUSHED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	KEYSLOT_REVOKED			Enable or disable interrupt for event KEYSLOT_REVOKED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	KEYSLOT_ERROR			Enable or disable interrupt for event KEYSLOT_ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				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	R/W	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable

7.17.5.6 INTENSET

Address offset: 0x304

Enable interrupt

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

7.17.5.7 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.17.5.8 INTPEND

Address offset: 0x30C

Pending interrupts

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	KEYSLOT_PUSHED			Read pending status of interrupt for event KEYSLOT_PUSHED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B	R	KEYSLOT_REVOKED			Read pending status of interrupt for event KEYSLOT_REVOKED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
C	R	KEYSLOT_ERROR			Read pending status of interrupt for event KEYSLOT_ERROR
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

7.17.5.9 STATUS

Address offset: 0x40C

Status bits for KMU operation

This register is reset and re-written by the KMU whenever SELECTKEYSLOT is written

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	SELECTED			Key slot ID successfully selected by the KMU
			Disabled	0	No key slot ID selected by KMU
			Enabled	1	Key slot ID successfully selected by KMU
B	R	BLOCKED			Violation status
			Disabled	0	No access violation detected
			Enabled	1	Access violation detected and blocked

7.17.5.10 SELECTKEYSLOT

Address offset: 0x500

Select key slot to be read over AHB or pushed over secure APB when TASKS_PUSH_KEYSLOT is started

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ID			Select key slot ID to be read over AHB, or pushed over secure APB, when TASKS_PUSH_KEYSLOT is started. NOTE: ID=0 is not a valid key slot ID. The 0 ID should be used when the KMU is idle or not in use. NOTE: Index N in UICR->KEYSLOT.KEY[N] and UICR->KEYSLOT.CONFIG[N] corresponds to KMU key slot ID=N+1.

7.18 LPCOMP — Low-power comparator

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
- Can be used as a wakeup source from System 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.

Note: LPCOMP cannot be used (STARTed) at the same time as COMP. Only one comparator can be used at a time.

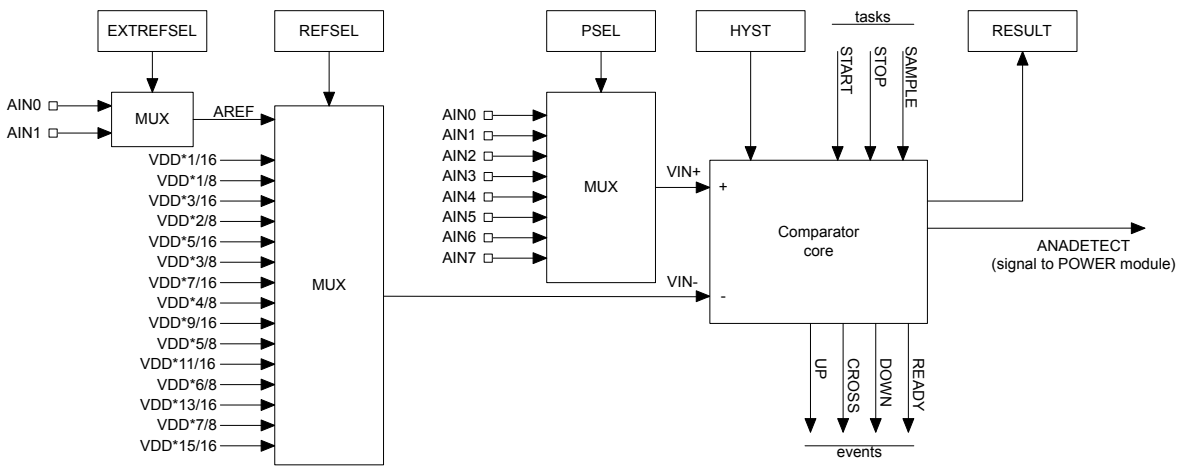

Figure 87: Low-power comparator

The wakeup comparator (LPCOMP) compares an input voltage (VIN+), which comes from an analog input pin selected via the PSEL register, against a reference voltage (VIN-) selected via registers REFSEL on page 288 and EXTREFSEL.

The PSEL, REFSEL, and EXTREFSEL registers must be configured before the LPCOMP is enabled through the ENABLE register.

The HYST register allows enabling an optional hysteresis in the comparator core. This hysteresis shall prevent noise on the signal to create unwanted events. Figure below illustrates 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.

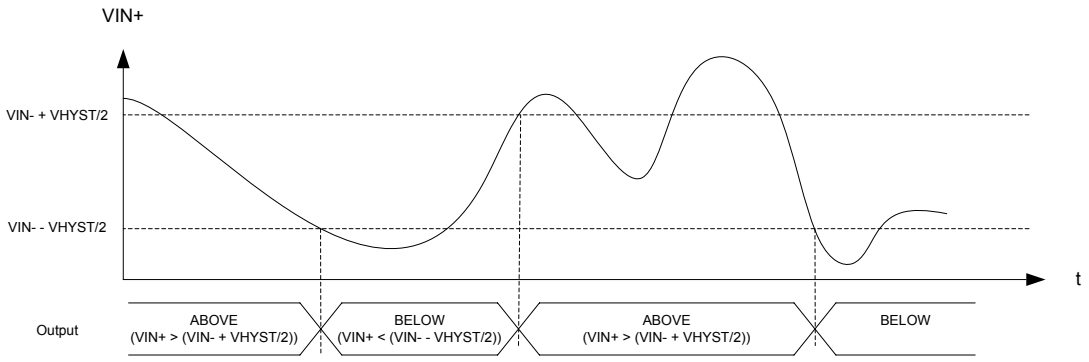

Figure 88: Effect of hysteresis on a noisy input signal

The LPCOMP is started by triggering the START task. After a startup time of $t_{LPCOMP,STARTUP}$, the LPCOMP will generate a READY event to indicate that the comparator is ready to use and the output of the LPCOMP is correct. The LPCOMP will generate events every time VIN+ crosses VIN-. More specifically, every time VIN+ rises above VIN- (upward crossing) an UP event is generated along with a CROSS event. Every time VIN+ falls below VIN- (downward crossing), a DOWN event is generated along with a CROSS event. When hysteresis is enabled, the upward crossing level becomes (VIN- + VHYST/2), and the downward crossing level becomes (VIN- - VHYST/2).

The LPCOMP is stopped by triggering the STOP task.

LPCOMP will be operational in both System ON and System OFF mode when it is enabled through the ENABLE register. See POWER — Power control on page 43 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 289) for more information on how to configure the ANADETECT signal.

The immediate value of the LPCOMP can be sampled to RESULT on page 288 by triggering the SAMPLE task.

See RESETREAS on page 66 for more information on how to detect a wakeup from LPCOMP.

7.18.1 Shared resources

The LPCOMP shares analog resources with SAADC. While it is possible to use the SAADC at the same time as the LPCOMP, selecting the same analog input pin for both modules is not supported.

Additionally, LPCOMP shares registers and other resources with other peripherals that have the same ID as the LPCOMP. See Peripherals with shared ID on page 147 for more information.

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 behavior.

7.18.2 Pin configuration

You can use the LPCOMP.PSEL register to select one of the analog input pins, **AIN0** through **AIN7**, as the analog input pin for the LPCOMP.

See GPIO — General purpose input/output on page 219 for more information about the pins. Similarly, you can use EXTREFSEL on page 289 to select one of the analog reference input pins, **AIN0** and **AIN1**, as input for AREF in case AREF is selected in EXTREFSEL on page 289. The selected analog pins will be acquired by the LPCOMP when it is enabled through ENABLE on page 288.

7.18.3 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5001A000 0x4001A000	APPLICATION	LPCOMP	LPCOMP : S LPCOMP : NS	US	NA	Low-power comparator	

Table 95: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start comparator
TASKS_STOP	0x004		Stop comparator
TASKS_SAMPLE	0x008		Sample comparator value
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
SUBSCRIBE_SAMPLE	0x088		Subscribe configuration for task SAMPLE
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
PUBLISH_READY	0x180		Publish configuration for event READY
PUBLISH_DOWN	0x184		Publish configuration for event DOWN
PUBLISH_UP	0x188		Publish configuration for event UP
PUBLISH_CROSS	0x18C		Publish configuration for event CROSS
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
RESULT	0x400		Compare result
ENABLE	0x500		Enable LPCOMP
PSEL	0x504		Input pin select
REFSEL	0x508		Reference select
EXTREFSEL	0x50C		External reference select
ANADETECT	0x520		Analog detect configuration
HYST	0x538		Comparator hysteresis enable

Table 96: Register overview

7.18.3.1 TASKS_START

Address offset: 0x000

Start comparator

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start comparator
			Trigger	1	Trigger task

7.18.3.2 TASKS_STOP

Address offset: 0x004

Stop comparator

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop comparator
			Trigger	1	Trigger task

7.18.3.3 TASKS_SAMPLE

Address offset: 0x008

Sample comparator 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_SAMPLE			Sample comparator value
			Trigger	1	Trigger task

7.18.3.4 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.18.3.5 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.18.3.6 SUBSCRIBE_SAMPLE

Address offset: 0x088

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SAMPLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.18.3.7 EVENTS_READY

Address offset: 0x100

LPCOMP is ready and output is valid

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_READY			LPCOMP is ready and output is valid
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.18.3.8 EVENTS_DOWN

Address offset: 0x104

Downward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DOWN			Downward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.18.3.9 EVENTS_UP

Address offset: 0x108

Upward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_UP			Upward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.18.3.10 EVENTS_CROSS

Address offset: 0x10C

Downward or upward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CROSS			Downward or upward crossing
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.18.3.11 PUBLISH_READY

Address offset: 0x180

Publish configuration for event READY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.18.3.12 PUBLISH_DOWN

Address offset: 0x184

Publish configuration for event DOWN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DOWN will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.18.3.13 PUBLISH_UP

Address offset: 0x188

Publish configuration for event UP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event UP will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.18.3.14 PUBLISH_CROSS

Address offset: 0x18C

Publish configuration for event CROSS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CROSS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.18.3.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY_SAMPLE			Shortcut between event READY and task SAMPLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	READY_STOP			Shortcut between event READY and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	DOWN_STOP			Shortcut between event DOWN and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	UP_STOP			Shortcut between event UP and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	CROSS_STOP			Shortcut between event CROSS and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.18.3.16 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for event READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to enable interrupt for event DOWN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to enable interrupt for event UP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to enable interrupt for event CROSS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.18.3.17 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	DOWN			Write '1' to disable interrupt for event DOWN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	UP			Write '1' to disable interrupt for event UP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	CROSS			Write '1' to disable interrupt for event CROSS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.18.3.18 RESULT

Address offset: 0x400

Compare result

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

ID	R/W	Field	Value ID	Value	Description
A	R	RESULT			Result of last compare. Decision point SAMPLE task.
			Below	0	Input voltage is below the reference threshold (VIN+ < VIN-)
			Above	1	Input voltage is above the reference threshold (VIN+ > VIN-)

7.18.3.19 ENABLE

Address offset: 0x500

Enable LPCOMP

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable LPCOMP
			Disabled	0	Disable
			Enabled	1	Enable

7.18.3.20 PSEL

Address offset: 0x504

Input pin select

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

ID	R/W	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

7.18.3.21 REFSEL

Address offset: 0x508

Reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	REFSEL			Reference select
			Ref1_8Vdd	0	VDD * 1/8 selected as reference
			Ref2_8Vdd	1	VDD * 2/8 selected as reference
			Ref3_8Vdd	2	VDD * 3/8 selected as reference
			Ref4_8Vdd	3	VDD * 4/8 selected as reference
			Ref5_8Vdd	4	VDD * 5/8 selected as reference
			Ref6_8Vdd	5	VDD * 6/8 selected as reference
			Ref7_8Vdd	6	VDD * 7/8 selected as reference
			ARef	7	External analog reference selected
			Ref1_16Vdd	8	VDD * 1/16 selected as reference
			Ref3_16Vdd	9	VDD * 3/16 selected as reference
			Ref5_16Vdd	10	VDD * 5/16 selected as reference
			Ref7_16Vdd	11	VDD * 7/16 selected as reference
			Ref9_16Vdd	12	VDD * 9/16 selected as reference
			Ref11_16Vdd	13	VDD * 11/16 selected as reference
			Ref13_16Vdd	14	VDD * 13/16 selected as reference
			Ref15_16Vdd	15	VDD * 15/16 selected as reference

7.18.3.22 EXTREFSEL

Address offset: 0x50C

External reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EXTREFSEL			External analog reference select
			AnalogReference0	0	Use AIN0 as external analog reference
			AnalogReference1	1	Use AIN1 as external analog reference

7.18.3.23 ANADETECT

Address offset: 0x520

Analog detect configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ANADETECT			Analog detect configuration
			Cross	0	Generate ANADETECT on crossing, both upward crossing and downward crossing
			Up	1	Generate ANADETECT on upward crossing only
			Down	2	Generate ANADETECT on downward crossing only

7.18.3.24 HYST

Address offset: 0x538

Comparator hysteresis enable

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

ID	R/W	Field	Value ID	Value	Description
A	RW	HYST			Comparator hysteresis enable
			Disabled	0	Comparator hysteresis disabled
			Enabled	1	Comparator hysteresis enabled

7.18.4 Electrical specification

7.18.4.1 LPCOMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{LPCANADET}$	Time from VIN crossing (≥ 50 mV above threshold) to ANADETECT signal generated		2.7		µs
$V_{INPOFFSET}$	Input offset including reference ladder error	-40		40	mV
V_{HYST}	Optional hysteresis		38		mV
$t_{STARTUP}$	Startup time for LPCOMP	28	51	134	µs

7.19 MUTEX — Mutual exclusive peripheral

The MUTEX peripheral uses mutual exclusion to support locking a resource that is shared between different CPUs in the system. The shared resource can only be used by one of these cores during the duration that it is locked.

The MUTEX peripheral includes several mutex registers. Each mutex register contains one bit which indicates if it is in a locked or unlocked state. Reading or writing to a mutex register may impact its state.

When a mutex is read, the following conditions apply:

- If the state is locked, the MUTEX[i] state is unchanged (remains in a locked state) and reading the register will return '1'.
- If the state is unlocked, the MUTEX[i] state changes to the locked state and reading the register will return '0'.

When writing '0' to a mutex, the following occurs:

- If the state is unlocked, the MUTEX[i] state is unchanged (remains in unlocked state) and the store is ignored.
- If the state is locked, the MUTEX[i] state changes to the unlocked state.

Note: Faults are not managed by the peripheral. If a mutex is locked and a fault occurs, it is the responsibility of the fault handler to release the mutex. If a fault handler is not managing the mutex release, the mutex will stay locked.

The following figure illustrates the mutex state transitions.

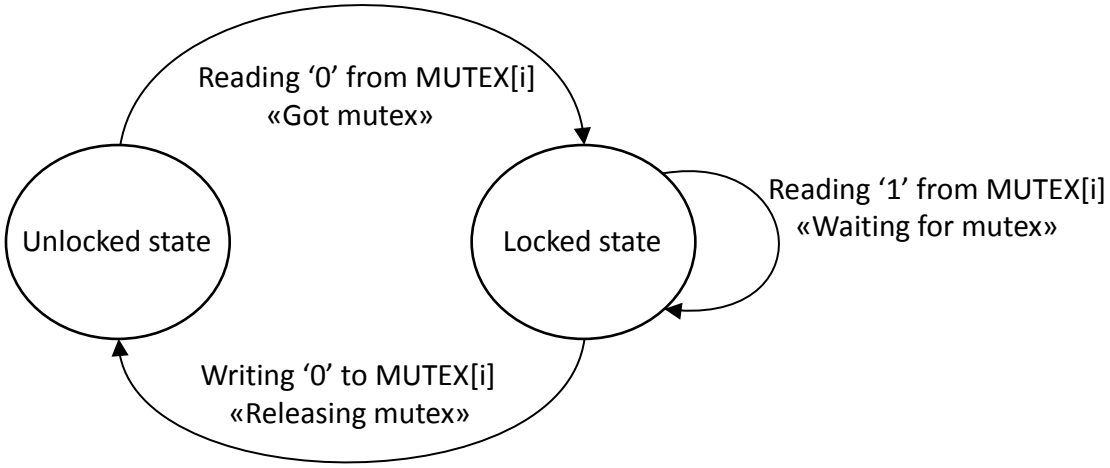

Figure 89: MUTEX - state transitions

The following code is an example of how a mutex can be used by two different CPUs:

```
// CPU0 Polling the MUTEX[0] register
while (NRF_MUTEX->MUTEX[0]){ // Read MUTEX[0]
                             // If a 0 is read, the mutex will lock
                             // If a 1 is read, polling will continue
}
// CPU0 working on shared data related to the mutex
 ...
NRF_MUTEX->MUTEX[0] = 0;  // Release the mutex
```

```
// CPU1 Polling the MUTEX[0] register
while (NRF_MUTEX->MUTEX[0]){ // Read MUTEX[0]
                             // If a 0 is read, the mutex will lock
                             // If a 1 is read, polling will continue
}
// CPU1 working on shared data related to the mutex
 ...
NRF_MUTEX->MUTEX[0] = 0;  // Release the mutex
```

Only one CPU can access the mutex at a time, meaning the mutex must be released before being accessed by the another CPU. If the load operation occurs at the same time, a bus arbitration mechanism will ensure only one CPU gets the mutex.

7.19.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50030000 0x40030000	APPLICATION	MUTEX	MUTEX : S MUTEX : NS	US	NA	Mutual exclusive hardware support	
0x50030000 0x40030000	NETWORK	MUTEX	APPMUTEX : S APPMUTEX : NS	US	NA	Mutex control	

Table 97: Instances

Register	Offset	Security	Description
MUTEX[n]	0x400		Mutex register

Table 98: Register overview

7.19.1.1 MUTEX[n] (n=0..15)

Address offset: 0x400 + (n × 0x4)

Mutex register

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																			A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value																															
A	RW	MUTEX																																	Mutex register n
			Unlocked	0																															Mutex n is in unlocked state
			Locked	1																															Mutex n is in locked state

7.20 NFCT — Near field communication tag

The NFCT peripheral is an implementation of an NFC Forum compliant listening device NFC-A.

With appropriate software, the NFCT peripheral can be used as the listening device NFC-A as specified by the NFC Forum.

Listed here are the main features for the NFCT peripheral:

- NFC-A listen mode operation
  - 13.56 MHz input frequency
  - Bit rate 106 kbps
- Wake-on-field low power field detection (SENSE) mode
- Frame assemble and disassemble for the NFC-A frames specified by the NFC Forum
- Programmable frame timing controller
- Integrated automatic collision resolution, cyclic redundancy check (CRC), and parity functions

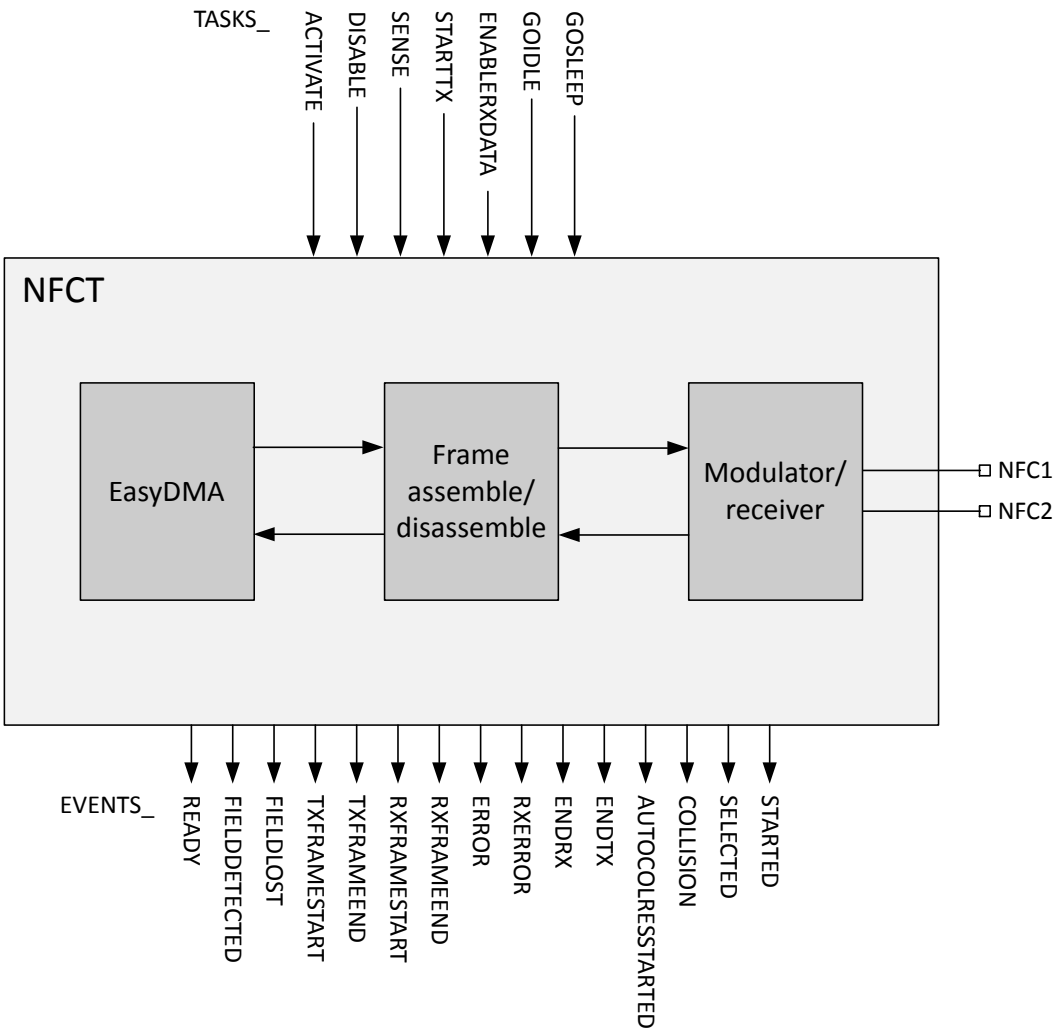

Figure 90: NFCT block diagram

7.20.1 Overview

The NFCT peripheral contains a 13.56 MHz AM receiver and a 13.56 MHz load modulator with 106 kbps data rate as defined by the NFC Forum.

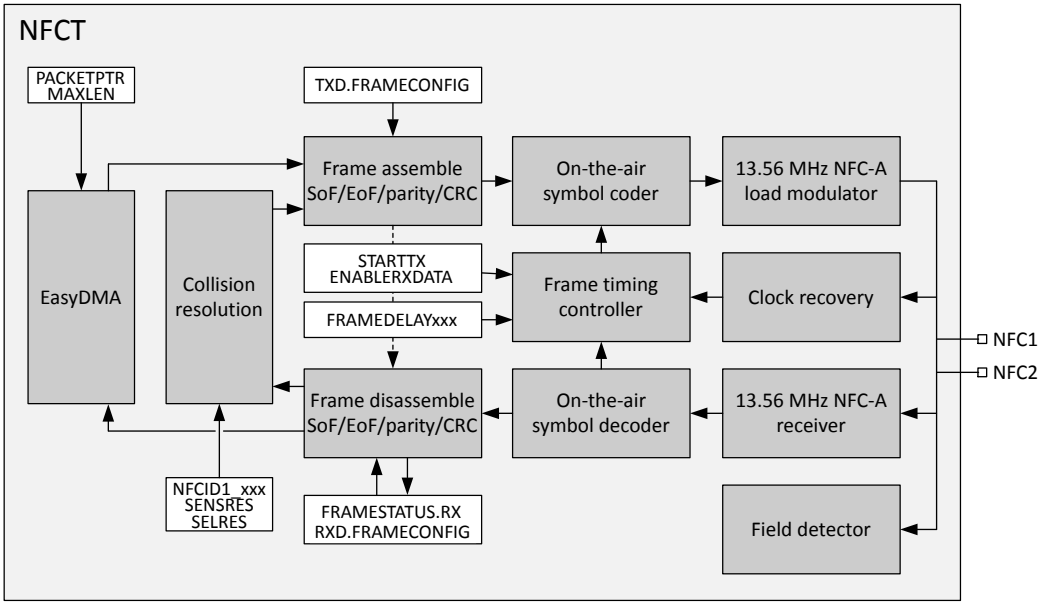

Figure 91: NFCT overview

When transmitting, the frame data will be transferred directly from RAM and transmitted with configurable frame type and delay timing. The system will be notified by an event whenever a complete frame is received or sent. The received frames will be automatically disassembled and the data part of the frame transferred to RAM.

The NFCT peripheral also supports the collision detection and resolution ("anticollision") as defined by the NFC Forum.

Wake-on-field is supported in SENSE mode while the device is either in System OFF or System ON mode. When the antenna enters an NFC field, an event will be triggered notifying the system to activate the NFCT functionality for incoming frames. In System ON, if the energy detected at the antenna increases beyond a threshold value, the module will generate a FIELDDETECTED event. When the strength of the field no longer supports NFC communication, the module will generate a FIELDLOST event. For the Low Power Field Detect threshold values, refer to NFCT Electrical Specification on page 329.

In System OFF, the NFCT Low Power Field Detect function can wake the system up through a reset. See RESETREAS on page 66 for more information on how to detect a wakeup from NFCT.

If the system is put into System OFF mode while a field is already present, the NFCT Low Power Field Detect function will wake the system up right away and generate a reset.

Note: As a consequence of a reset, NFCT is disabled, and therefore the reset handler will have to activate NFCT again and set it up properly.

The HFXO must be running before the NFCT peripheral goes into ACTIVATED state. Note that the NFCT peripheral calibration is automatically done on ACTIVATE task. The HFXO can be turned off when the NFCT peripheral goes into SENSE mode. The shortcut FIELDDETECTED_ACTIVATE can be used when the HFXO is already running while in SENSE mode.

Outgoing data will be collected from RAM with the EasyDMA function and assembled according to 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 NFCT peripheral includes a frame timing controller that can be used to accurately control the inter-frame delay between the incoming frame and a corresponding outgoing frame. It also includes optional CRC functionality.

7.20.2 Operating states

Tasks and events are used to control the operating state of the peripheral. The module can change state by triggering a task, or when specific operations are finalized. Events and tasks allow software to keep track of and change the current state.

See NFCT block diagram on page 293 and NFCT state diagram, automatic collision resolution enabled on page 295 for more information. See *NFC Forum, NFC Activity Technical Specification* for description on NFCT operating states.

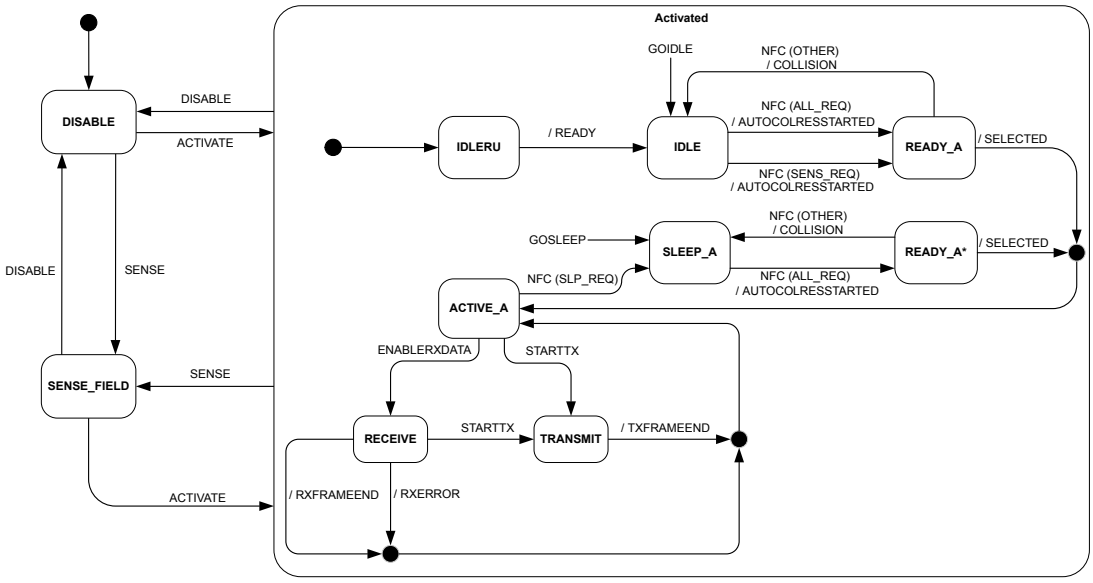

Figure 92: NFCT state diagram, automatic collision resolution enabled

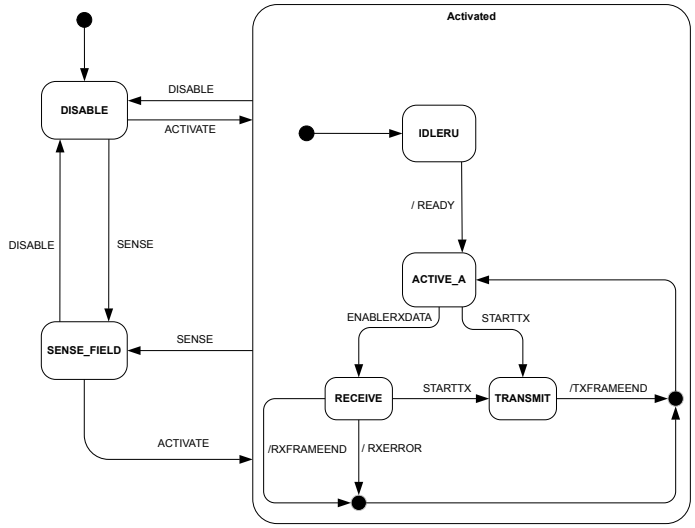

Figure 93: NFCT state diagram, automatic collision resolution disabled

Important:

- FIELDLOST event is not generated in SENSE mode.
- Sending SENSE task while field is still present does not generate FIELDDETECTED event.
- If the FIELDDETECTED event is cleared before sending the ACTIVATE task, then the FIELDDETECTED event shows up again after sending the ACTIVATE task. The shortcut FIELDDETECTED_ACTIVATE can be used to avoid this condition.

7.20.3 Pin configuration

NFCT uses two pins to connect the antenna and these pins are shared with GPIOs.

The PROTECT field in the NFCPINS register in UICR defines the usage of these pins and their protection level against excessive voltages. The content of the NFCPINS register is reloaded at every reset. See Pin assignments on page 783 for the pins used by the NFCT peripheral.

When NFCPINS.PROTECT=NFC, a protection circuit will be enabled on the dedicated pins, preventing the chip from being damaged in the presence of a strong NFC field. The protection circuit will short the two pins together if voltage difference exceeds approximately 2V. The GPIO function on those pins will also be disabled.

When NFCPINS.PROTECT=Disabled, the device will not be protected against strong NFC field damages caught by a connected NFCT antenna, and the NFCT peripheral will not operate as expected, as it will never leave the DISABLE state.

The pins dedicated to the NFCT antenna function will have some limitation when the pins are configured for normal GPIO operation. The pin capacitance will be higher on those (refer to C_{PAD_NFC} in the Electrical Specification of GPIO — General purpose input/output on page 219), and some increased leakage current between the two pins is to be expected if they are used in GPIO mode, and are driven to different logical values. To save power, the two pins should always be set to the same logical value whenever entering one of the device power saving modes. For details, refer to I_{NFC_LEAK} in the Electrical Specification of GPIO — General purpose input/output on page 219.

7.20.4 EasyDMA

The NFCT peripheral implements EasyDMA for reading and writing of data packets from and to the Data RAM.

The NFCT EasyDMA utilizes a pointer called PACKETPTR on page 324 for receiving and transmitting packets.

The NFCT peripheral uses EasyDMA to read or write RAM, but not both at the same time. The event RXFRAMESTART indicates that the EasyDMA has started writing to the RAM for a receive frame and the event RXFRAMEND indicates that the EasyDMA has completed writing to the RAM. Similarly, the event TXFRAMESTART indicates that the EasyDMA has started reading from the RAM for a transmit frame and the event TXFRAMEND indicates that the EasyDMA has completed reading from the RAM. If a transmit and a receive operation is issued at the same time, the transmit operation would be prioritized.

Starting a transmit operation while the EasyDMA is writing a receive frame to the RAM will result in unpredictable behavior. Starting an EasyDMA operation when there is an ongoing EasyDMA operation may result in unpredictable behavior. It is recommended to wait for the TXFRAMEEND or RXFRAMEEND event for the ongoing transmit or receive before starting a new receive or transmit operation.

The MAXLEN on page 324 register determines the maximum number of bytes that can be read from or written to the RAM. This feature can be used to ensure that the NFCT peripheral does not overwrite, or read beyond, the RAM assigned to a packet. Note that if the RXD.AMOUNT or TXD.AMOUNT register indicates longer data packets than set in MAXLEN, the frames sent to or received from the physical layer will be incomplete. If that situation occurs in RX mode, the OVERRUN bit in the FRAMESTATUS.RX register will be set and an RXERROR event will be triggered.

Important: The RXD.AMOUNT and TXD.AMOUNT define a frame length in bytes and bits excluding start of frame (SoF), end of frame (EoF), and parity, but including CRC for RXD.AMOUNT only. Make sure to take potential additional bits into account when setting MAXLEN.

Only sending task ENABLERXDATA ensures that a new value in PACKETPTR pointing to the RX buffer in Data RAM is taken into account.

If PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a hard fault or RAM corruption. For more information about the different memory regions, see Chapter Memory on page 18.

The NFCT peripherals normally do alternative receive and transmit frames. Therefore, to prepare for the next frame, the PACKETPTR, MAXLEN, TXD.FRAMECONFIG and TXD.AMOUNT can be updated while the receive is in progress, and, similarly, the PACKETPTR, MAXLEN and RXD.FRAMECONFIG can be updated while the transmit is in progress. They can be updated and prepared for the next NFC frame immediately after the STARTED event of the current frame has been received. Updating the TXD.FRAMECONFIG and TXD.AMOUNT during the current transmit frame or updating RXD.FRAMECONFIG during current receive frame may cause unpredictable behaviour.

In accordance with *NFC Forum, NFC Digital Protocol Technical Specification*, the least significant bit (LSB) from the least significant byte (LSByte) is sent on air first. The bytes are stored in increasing order, starting at the lowest address in the EasyDMA buffer in RAM.

7.20.5 Frame assembler

The NFCT peripheral implements a frame assembler in hardware.

When the NFCT peripheral is in the ACTIVE_A state, the software can decide to enter RX or TX mode. For RX mode, see Frame disassembler on page 298. For TX mode, the software must indicate the address of the source buffer in Data RAM and its size through programming the PACKETPTR and MAXLEN registers respectively, then issuing a STARTTX task.

MAXLEN must be set so that it matches the size of the frame to be sent.

The STARTED event indicates that the PACKETPTR and MAXLEN registers have been captured by the frame assembler EasyDMA.

When asserting the STARTTX task, the frame assembler module will start reading TXD.AMOUNT.TXDATABYTES bytes (plus one additional byte if TXD.AMOUNT.TXDATABITS > 0) from the RAM position set by the PACKETPTR.

The NFCT peripheral transmits the data as read from RAM, adding framing and the CRC calculated on the fly if set in TXD.FRAMECONFIG. The NFCT peripheral will take (8*TXD.AMOUNT.TXDATABYTES + TXD.AMOUNT.TXDATABITS) bits and assemble a frame according to the settings in TXD.FRAMECONFIG. Both short frames, standard frames, and bit-oriented SDD frames as specified in the *NFC Forum, NFC Digital Protocol Technical Specification* can be assembled by the correct setting of the TXD.FRAMECONFIG register.

The bytes will be transmitted on air in the same order as they are read from RAM with a rising bit order within each byte, least significant bit (LSB) first. That is, the least significant bit (b0) will be transmitted on air before the second bit (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*.

Note: Some NFC Forum documents, such as *NFC Forum, NFC Digital Protocol Technical Specification*, define bit numbering in a byte from b1 (LSB) to b8 (most significant bit (MSB)), while most other technical documents from the NFC Forum, and also the Nordic Semiconductor documentation, traditionally number them from b0 to b7. The present document uses the b0–b7 numbering scheme. Be aware of this when comparing the *NFC Forum, NFC Digital Protocol Technical Specification* to others.

The frame assembler can be configured in TXD.FRAMECONFIG to add SoF symbol, calculate and add parity bits, and calculate and add CRC to the data read from RAM when assembling the frame. The total frame will then be longer than what is defined by TXD.AMOUNT.TXDATABYTES. TXDATABITS. DISCARDMODE will select if the first bits in the first byte read from RAM or the last bits in the last byte read from RAM will be discarded if TXD.AMOUNT.TXDATABITS are not equal to zero. Note that if TXD.FRAMECONFIG.PARITY

= Parity and TXD.FRAMECONFIG.DISCARDMODE=DiscardStart, a parity bit will be included after the non-complete first byte. No parity will be added after a non-complete last byte.

The frame assemble operation for different settings in TXD.FRAMECONFIG is illustrated in the following table. All shaded bit fields are added by the frame assembler. Some of these bits are optional and appearances are configured in TXD.FRAMECONFIG. Note that the frames illustrated do not necessarily comply with the NFC specification. The figure only illustrates the behavior of the NFCT peripheral.

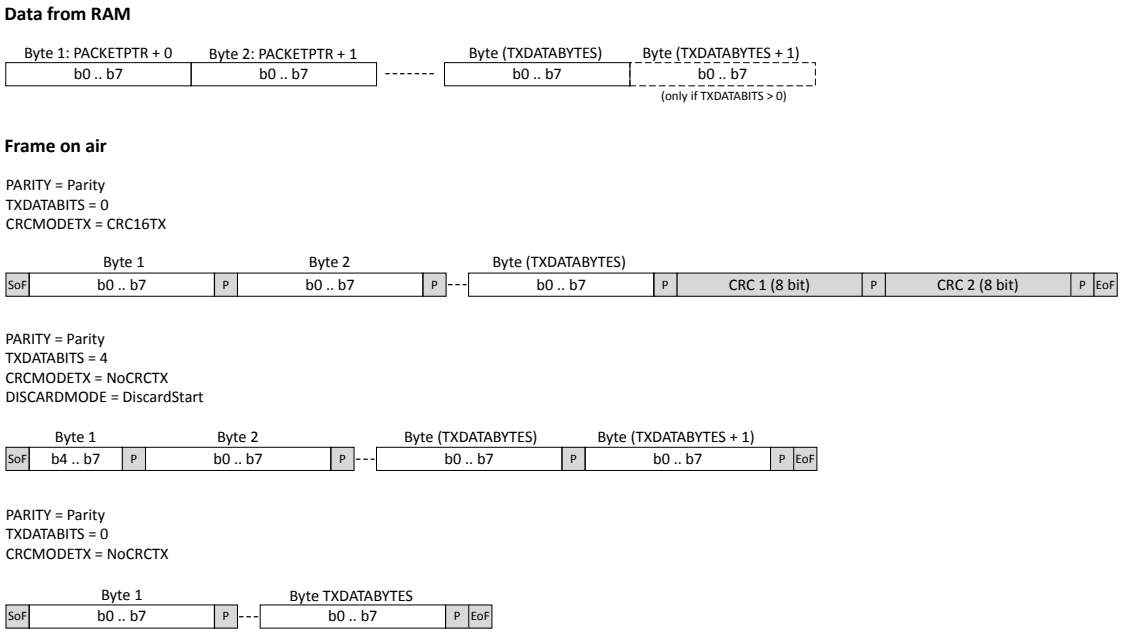

Figure 94: Frame assemble illustration

The accurate timing for transmitting the frame on air is set using the frame timing controller settings.

7.20.6 Frame disassembler

The NFCT peripheral implements a frame disassembler in hardware.

When the NFCT peripheral is in the ACTIVE_A state, the software can decide to enter RX or TX mode. For TX mode, see Frame assembler on page 297. For RX mode, the software must indicate the address and size of the destination buffer in Data RAM through programming the PACKETPTR and MAXLEN registers before issuing an ENABLERXDATA task.

The STARTED event indicates that the PACKETPTR and MAXLEN registers have been captured by the frame disassembler EasyDMA.

When an incoming frame starts, the RXFRAMESTART event will get issued and data will be written to the buffer in Data RAM. The frame disassembler will verify and remove any parity bits, start of frame (SoF) and end of frame (EoF) symbols on the fly based on RXD.FRAMECONFIG register configuration. It will, however, verify and transfer the CRC bytes into RAM, if the CRC is enabled through RXD.FRAMECONFIG.

When an EoF symbol is detected, the NFCT peripheral will assert the RXFRAMEEND event and write the RXD.AMOUNT register to indicate numbers of received bytes and bits in the data packet. The module does not interpret the content of the data received from the remote NFC device, except for SoF, EoF, parity, and CRC checking, as described above. The frame disassemble operation is illustrated in the following figure.

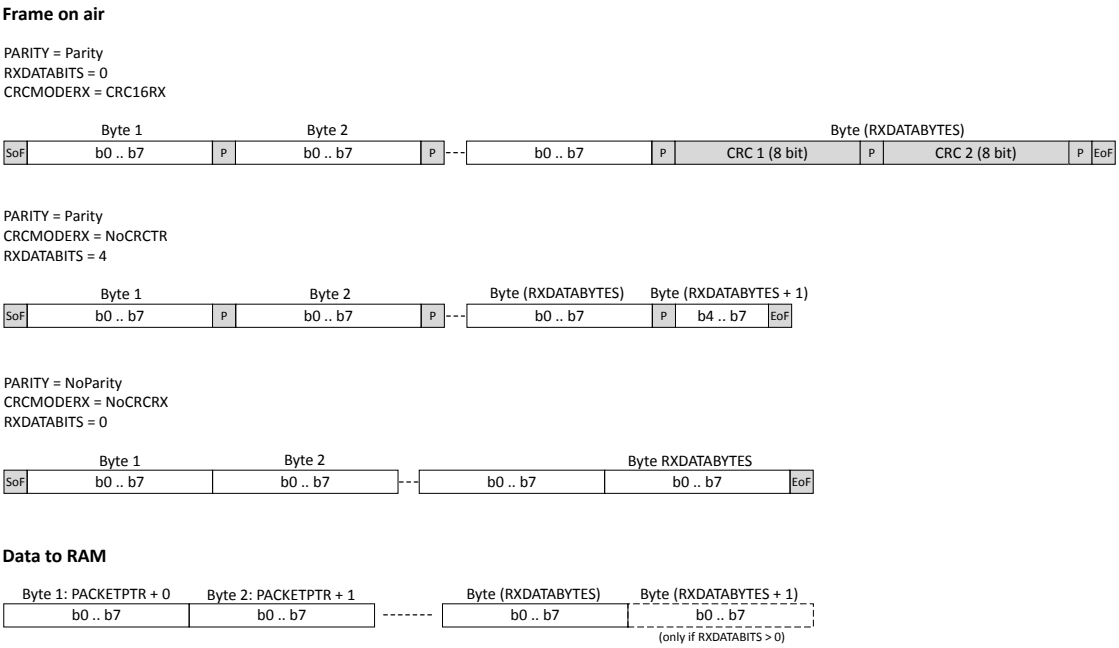

Figure 95: Frame disassemble illustration

Per NFC specification, the time between EoF to the next SoF can be as short as 86 µs, and thefore care must be taken that PACKETPTR and MAXLEN are ready and ENABLERXDATA is issued on time after the end of previous frame. The use of a PPI shortcut from TXFRAMEEND to ENABLERXDATA is recommended.

7.20.7 Frame timing controller

The NFCT peripheral includes a frame timing controller that continuously keeps track of the number of the 13.56 MHz RF carrier clock periods since the end of the EoF of the last received frame.

The NFCT peripheral can be programmed to send a responding frame within a time window or at an exact count of RF carrier periods. In case of FRAMEDELAYMODE = Window, a STARTTX task triggered before the frame timing controller counter is equal to FRAMEDELAYMIN will force the transmission to halt until the counter is equal to FRAMEDELAYMIN. If the counter is within FRAMEDELAYMIN and FRAMEDELAYMAX when the STARTTX task is triggered, the NFCT peripheral will start the transmission straight away. In case of FRAMEDELAYMODE = ExactVal, a STARTTX task triggered before the frame delay counter is equal to FRAMEDELAYMAX will halt the actual transmission start until the counter is equal to FRAMEDELAYMAX.

In case of FRAMEDELAYMODE = WindowGrid, the behaviour is similar to the FRAMEDELAYMODE = Window, but the actual transmission between FRAMEDELAYMIN and FRAMEDELAYMAX starts on a bit grid as defined for NFC-A Listen frames (slot duration of 128 RF carrier periods).

An ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) will be asserted if the frame timing controller counter reaches FRAMEDELAYMAX without any STARTTX task triggered. This may happen even when the response is not required as per *NFC Forum, NFC Digital Protocol Technical Specification*. Any commands handled by the automatic collision resolution that don't involve a response being generated may also result in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS). The FRAMEDELAYMIN and FRAMEDELAYMAX values shall only be updated before the STARTTX task is triggered. Failing to do so may cause unpredictable behaviour.

The frame timing controller operation is illustrated in the following figure. 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*.

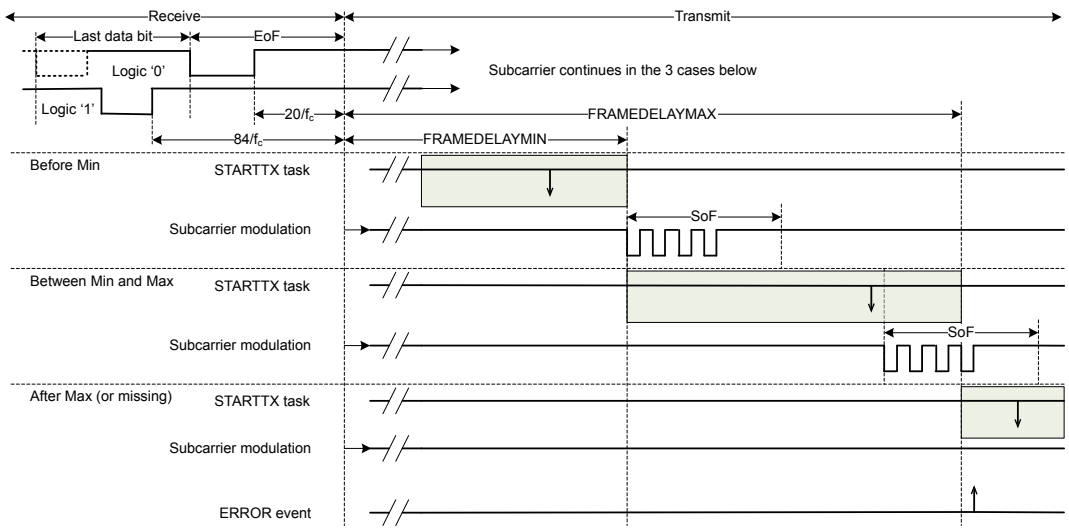

Figure 96: Frame timing controller (FRAMEDELAYMODE=Window)

7.20.8 Collision resolution

The NFCT peripheral implements an automatic collision resolution function as defined by the NFC Forum.

Automatic collision resolution is enabled by default, and it is recommended that the feature is used since it is power efficient and reduces the complexity of software handling the collision resolution sequence. This feature can be disabled through the MODE field in the AUTOCOLRESCONFIG register. When the automatic collision resolution is disabled, all commands will be sent over EasyDMA as defined in frame disassembler.

The SENSRES and SELRES registers need to be programmed upfront in order for the collision resolution to behave correctly. Depending on the NFCIDSIZE field in SENSRES, the following registers also need to be programmed upfront:

- NFCID1_LAST if NFCID1SIZE=NFCID1Single (ID = 4 bytes);
- NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Double (ID = 7 bytes);
- NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Triple (ID = 10 bytes);

A pre-defined set of registers, NFC.TAGHEADER0..3, containing a valid NFCID1 value, is available in FICR and can be used by software to populate the NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST registers.

NFCID1 byte allocation (top sent first on air) on page 301 explains the position of the ID bytes in NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST, depending on the ID size, and as compared to the definition used in the *NFC Forum, NFC Digital Protocol Technical Specification*.

	ID = 4 bytes	ID = 7 bytes	ID = 10 bytes
NFCID1_Q			$nfcid1_0$
NFCID1_R			$nfcid1_1$
NFCID1_S			$nfcid1_2$
NFCID1_T		$nfcid1_0$	$nfcid1_3$
NFCID1_U		$nfcid1_1$	$nfcid1_4$
NFCID1_V		$nfcid1_2$	$nfcid1_5$
NFCID1_W	$nfcid1_0$	$nfcid1_3$	$nfcid1_6$
NFCID1_X	$nfcid1_1$	$nfcid1_4$	$nfcid1_7$
NFCID1_Y	$nfcid1_2$	$nfcid1_5$	$nfcid1_8$
NFCID1_Z	$nfcid1_3$	$nfcid1_6$	$nfcid1_9$

Table 99: NFCID1 byte allocation (top sent first on air)

The hardware implementation can handle the states from IDLE to ACTIVE_A automatically as defined in the *NFC Forum, NFC Activity Technical Specification*, and the other states are to be handled by software. The software keeps track of the state through events. The collision resolution will trigger an AUTOCOLRESSTARTED event when it has started. Reaching the ACTIVE_A state is indicated by the SELECTED event.

If collision resolution fails, a COLLISION event is triggered. Note that errors occurring during automatic collision resolution may also cause ERROR and/or RXERROR events to be generated. Other events may also get generated. It is recommended that the software ignores any event except COLLISION, SELECTED and FIELDLOST during automatic collision resolution. Software shall also make sure that any unwanted SHORT or PPI shortcut is disabled during automatic collision resolution.

The automatic collision resolution will be restarted, if the packets are received with CRC or parity errors while in ACTIVE_A state. The automatic collision resolution feature can be disabled while in ACTIVE_A state to avoid this.

The SLP_REQ is automatically handled by the NFCT peripheral when the automatic collision resolution is enabled. However, this results in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) since the SLP_REQ has no response. This error must be ignored until the SELECTED event is triggered and this error should be cleared by the software when the SELECTED event is triggered.

7.20.9 Antenna interface

In ACTIVATED state, an amplitude regulator will adjust the voltage swing on the antenna pins to a value that is within the V_{swing} limit.

Refer to NFCT Electrical Specification on page 329.

7.20.10 NFCT antenna recommendations

The NFCT antenna coil must be connected differential between **NFC1** and **NFC2** pins of the device.

Two external capacitors should be used to tune the resonance of the antenna circuit to 13.56 MHz.

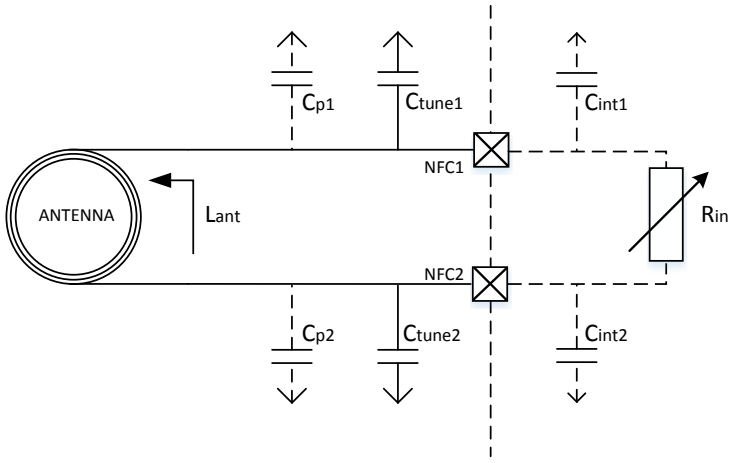

Figure 97: NFCT antenna recommendations

The required tuning capacitor value is given by the below equations:

$$C'_{tune} = \frac{1}{(2\pi \cdot 13.56\ MHz)^2 \cdot L_{ant}} \quad where\ C'_{tune} = \frac{1}{2} \cdot (C_p + C_{\text{int}} + C_{tune})$$

$$and\ C_{tune1} = C_{tune2} = C_{tune} \qquad C_{p1} = C_{p2} = C_p \qquad C_{\text{int1}} = C_{\text{int2}} = C_{\text{int}}$$

$$C_{tune} = \frac{2}{(2\pi \cdot 13.56\ MHz)^2 \cdot L_{ant}} - C_p - C_{\text{int}}$$

An antenna inductance of L_{ant} = 2 µH will give tuning capacitors in the range of 130 pF on each pin. The total capacitance on **NFC1** and **NFC2** must be matched.

7.20.11 Battery protection

If the antenna is exposed to a strong NFC field, current may flow in the opposite direction on the supply due to parasitic diodes and ESD structures.

If the battery used does not tolerate return current, a series diode must be placed between the battery and the device in order to protect the battery.

7.20.12 Digital Modulation Signal

Support for external analog frontends or antenna architectures is possible by optionally outputting the digital modulation signal to a GPIO.

The NFCT peripheral is designed to connect directly to a loop antenna, receive a modulated signal from an NFC Reader with its internal analog frontend and transmit data back by changing the input resistance that is then seen as modulated load by the NFC Reader.

In addition, the peripheral has an option to output the digital modulation signal to a GPIO. Reception still occurs through the internal analog frontend, whereas transmission can be done by one of the following:

- The internal analog frontend through the loop antenna (default)
- An external frontend using the digital modulation signal
- The combination of both above

There are two registers that allow configuration of the modulation signal (i.e. of the response from NFCT to the NFC Reader), MODULATIONCTRL and MODULATIONPSEL. The registers need to be programmed before NFCT sends a response to a request from a reader. Ideally, this configuration is performed during startup and whenever the NFCT peripheral is powered up.

The selected GPIO needs to be configured as output in the corresponding GPIO configuration register. It is recommended to set an output value in the corresponding GPIO.OUT register – this value will be driven whenever the NFCT peripheral is disabled.

NFCT drives the pin low when there is no modulation, and drives it with On-Off Keying (OOK) modulation of an 847 kHz subcarrier (derived from the carrier frequency) when it responds to commands from an NFC Reader.

7.20.13 References

NFC Forum, NFC Analog Specification version 2.1, www.nfc-forum.org

NFC Forum, NFC Digital Protocol Technical Specification version 2.2, www.nfc-forum.org

NFC Forum, NFC Activity Technical Specification version 2.1, www.nfc-forum.org

7.20.14 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5002D000 0x4002D000	APPLICATION	NFCT	NFCT : S NFCT : NS	US	SA	Near field communication tag	

Table 100: Instances

Register	Offset	Security	Description
TASKS_ACTIVATE	0x000		Activate NFCT peripheral for incoming and outgoing frames, change state to activated
TASKS_DISABLE	0x004		Disable NFCT peripheral
TASKS_SENSE	0x008		Enable NFC sense field mode, change state to sense mode
TASKS_STARTTX	0x00C		Start transmission of an outgoing frame, change state to transmit
TASKS_ENABLERXDATA	0x01C		Initializes the EasyDMA for receive.
TASKS_GOIDLE	0x024		Force state machine to IDLE state
TASKS_GOSLEEP	0x028		Force state machine to SLEEP_A state
SUBSCRIBE_ACTIVATE	0x080		Subscribe configuration for task ACTIVATE
SUBSCRIBE_DISABLE	0x084		Subscribe configuration for task DISABLE
SUBSCRIBE_SENSE	0x088		Subscribe configuration for task SENSE
SUBSCRIBE_STARTTX	0x08C		Subscribe configuration for task STARTTX
SUBSCRIBE_ENABLERXDATA	0x09C		Subscribe configuration for task ENABLERXDATA
SUBSCRIBE_GOIDLE	0x0A4		Subscribe configuration for task GOIDLE
SUBSCRIBE_GOSLEEP	0x0A8		Subscribe configuration for task GOSLEEP
EVENTS_READY	0x100		The NFCT peripheral is ready to receive and send frames
EVENTS_FIELDDETECTED	0x104		Remote NFC field detected
EVENTS_FIELDLOST	0x108		Remote NFC field lost
EVENTS_TXFRAMESTART	0x10C		Marks the start of the first symbol of a transmitted frame
EVENTS_TXFRAMEEND	0x110		Marks the end of the last transmitted on-air symbol of a frame
EVENTS_RXFRAMESTART	0x114		Marks the end of the first symbol of a received frame
EVENTS_RXFRAMEEND	0x118		Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer
EVENTS_ERROR	0x11C		NFC error reported. The ERRORSTATUS register contains details on the source of the error.

Register	Offset	Security	Description
EVENTS_RXERROR	0x128		NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.
EVENTS_ENDRX	0x12C		RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.
EVENTS_ENDTX	0x130		Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer
EVENTS_AUTOCOLRESSTARTED	0x138		Auto collision resolution process has started
EVENTS_COLLISION	0x148		NFC auto collision resolution error reported.
EVENTS_SELECTED	0x14C		NFC auto collision resolution successfully completed
EVENTS_STARTED	0x150		EasyDMA is ready to receive or send frames.
PUBLISH_READY	0x180		Publish configuration for event READY
PUBLISH_FIELDDETECTED	0x184		Publish configuration for event FIELDDETECTED
PUBLISH_FIELDLOST	0x188		Publish configuration for event FIELDLOST
PUBLISH_TXFRAMESTART	0x18C		Publish configuration for event TXFRAMESTART
PUBLISH_TXFRAMEEND	0x190		Publish configuration for event TXFRAMEEND
PUBLISH_RXFRAMESTART	0x194		Publish configuration for event RXFRAMESTART
PUBLISH_RXFRAMEEND	0x198		Publish configuration for event RXFRAMEEND
PUBLISH_ERROR	0x19C		Publish configuration for event ERROR
PUBLISH_RXERROR	0x1A8		Publish configuration for event RXERROR
PUBLISH_ENDRX	0x1AC		Publish configuration for event ENDRX
PUBLISH_ENDTX	0x1B0		Publish configuration for event ENDTX
PUBLISH_AUTOCOLRESSTARTED	0x1B8		Publish configuration for event AUTOCOLRESSTARTED
PUBLISH_COLLISION	0x1C8		Publish configuration for event COLLISION
PUBLISH_SELECTED	0x1CC		Publish configuration for event SELECTED
PUBLISH_STARTED	0x1D0		Publish configuration for event STARTED
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSTATUS	0x404		NFC Error Status register
FRAMESTATUS.RX	0x40C		Result of last incoming frame
NFCTAGSTATE	0x410		Current operating state of NFC tag
SLEEPSTATE	0x420		Sleep state during automatic collision resolution
FIELDPRESENT	0x43C		Indicates the presence or not of a valid field
FRAMEDELAYMIN	0x504		Minimum frame delay
FRAMEDELAYMAX	0x508		Maximum frame delay
FRAMEDELAYMODE	0x50C		Configuration register for the Frame Delay Timer
PACKETPTR	0x510		Packet pointer for TXD and RXD data storage in Data RAM
MAXLEN	0x514		Size of the RAM buffer allocated to TXD and RXD data storage each
TXD.FRAMECONFIG	0x518		Configuration of outgoing frames
TXD.AMOUNT	0x51C		Size of outgoing frame
RXD.FRAMECONFIG	0x520		Configuration of incoming frames
RXD.AMOUNT	0x524		Size of last incoming frame
MODULATIONCTRL	0x52C		Enables the modulation output to a GPIO pin which can be connected to a second external antenna.
MODULATIONPSEL	0x538		Pin select for Modulation control
NFCID1_LAST	0x590		Last NFCID1 part (4, 7 or 10 bytes ID)
NFCID1_2ND_LAST	0x594		Second last NFCID1 part (7 or 10 bytes ID)
NFCID1_3RD_LAST	0x598		Third last NFCID1 part (10 bytes ID)
AUTOCOLRESCONFIG	0x59C		Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is activated.
SENSRES	0x5A0		NFC-A SENS_RES auto-response settings
SELRES	0x5A4		NFC-A SEL_RES auto-response settings

Table 101: Register overview

7.20.14.1 TASKS_ACTIVATE

Address offset: 0x000

Activate NFCT peripheral for incoming and outgoing frames, change state to activated

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_ACTIVATE			Activate NFCT peripheral for incoming and outgoing frames, change state to activated
			Trigger	1	Trigger task

7.20.14.2 TASKS_DISABLE

Address offset: 0x004

Disable NFCT peripheral

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_DISABLE			Disable NFCT peripheral
			Trigger	1	Trigger task

7.20.14.3 TASKS_SENSE

Address offset: 0x008

Enable NFC sense field mode, change state to sense mode

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_SENSE			Enable NFC sense field mode, change state to sense mode
			Trigger	1	Trigger task

7.20.14.4 TASKS_STARTTX

Address offset: 0x00C

Start transmission of an outgoing frame, change state to transmit

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start transmission of an outgoing frame, change state to transmit
			Trigger	1	Trigger task

7.20.14.5 TASKS_ENABLERXDATA

Address offset: 0x01C

Initializes the EasyDMA for receive.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_ENABLERXDATA			Initializes the EasyDMA for receive.
			Trigger	1	Trigger task

7.20.14.6 TASKS_GOIDLE

Address offset: 0x024

Force state machine to IDLE state

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_GOIDLE			Force state machine to IDLE state
			Trigger	1	Trigger task

7.20.14.7 TASKS_GOSLEEP

Address offset: 0x028

Force state machine to SLEEP_A state

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_GOSLEEP			Force state machine to SLEEP_A state
			Trigger	1	Trigger task

7.20.14.8 SUBSCRIBE_ACTIVATE

Address offset: 0x080

Subscribe configuration for task ACTIVATE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task ACTIVATE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.9 SUBSCRIBE_DISABLE

Address offset: 0x084

Subscribe configuration for task 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task DISABLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.10 SUBSCRIBE_SENSE

Address offset: 0x088

Subscribe configuration for task SENSE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SENSE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.11 SUBSCRIBE_STARTTX

Address offset: 0x08C

Subscribe configuration for task STARTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.12 SUBSCRIBE_ENABLERXDATA

Address offset: 0x09C

Subscribe configuration for task ENABLERXDATA

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task ENABLERXDATA will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.13 SUBSCRIBE_GOIDLE

Address offset: 0x0A4

Subscribe configuration for task GOIDLE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task GOIDLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.14 SUBSCRIBE_GOSLEEP

Address offset: 0x0A8

Subscribe configuration for task GOSLEEP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task GOSLEEP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.20.14.15 EVENTS_READY

Address offset: 0x100

The NFCT peripheral is ready to receive and send 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_READY			The NFCT peripheral is ready to receive and send frames
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.16 EVENTS_FIELDDETECTED

Address offset: 0x104

Remote NFC field detected

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_FIELDDETECTED			Remote NFC field detected
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.17 EVENTS_FIELDLOST

Address offset: 0x108

Remote NFC field lost

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_FIELDLOST			Remote NFC field lost
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.18 EVENTS_TXFRAMESTART

Address offset: 0x10C

Marks the start of the first symbol of a transmitted frame

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXFRAMESTART			Marks the start of the first symbol of a transmitted frame
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.19 EVENTS_TXFRAMEEND

Address offset: 0x110

Marks the end of the last transmitted on-air symbol of a frame

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXFRAMEEND			Marks the end of the last transmitted on-air symbol of a frame
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.20 EVENTS_RXFRAMESTART

Address offset: 0x114

Marks the end of the first symbol of a received frame

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXFRAMESTART			Marks the end of the first symbol of a received frame
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.21 EVENTS_RXFRAMEEND

Address offset: 0x118

Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXFRAMEEND			Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.22 EVENTS_ERROR

Address offset: 0x11C

NFC error reported. The ERRORSTATUS register contains details on the source of the error.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			NFC error reported. The ERRORSTATUS register contains details on the source of the error.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.23 EVENTS_RXERROR

Address offset: 0x128

NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXERROR			NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.24 EVENTS_ENDRX

Address offset: 0x12C

RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.25 EVENTS_ENDTX

Address offset: 0x130

Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX 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
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.26 EVENTS_AUTOCOLRESSTARTED

Address offset: 0x138

Auto collision resolution process has started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_AUTOCOLRESSTARTED			Auto collision resolution process has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.27 EVENTS_COLLISION

Address offset: 0x148

NFC auto collision resolution error reported.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_COLLISION			NFC auto collision resolution error reported.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.28 EVENTS_SELECTED

Address offset: 0x14C

NFC auto collision resolution successfully completed

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SELECTED			NFC auto collision resolution successfully completed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.29 EVENTS_STARTED

Address offset: 0x150

EasyDMA is ready to receive or send 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			EasyDMA is ready to receive or send frames.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.20.14.30 PUBLISH_READY

Address offset: 0x180

Publish configuration for event READY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.31 PUBLISH_FIELDDETECTED

Address offset: 0x184

Publish configuration for event FIELDDETECTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event FIELDDETECTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.32 PUBLISH_FIELDLOST

Address offset: 0x188

Publish configuration for event FIELDLOST

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event FIELDLOST will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.33 PUBLISH_TXFRAMESTART

Address offset: 0x18C

Publish configuration for event TXFRAMESTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXFRAMESTART will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.34 PUBLISH_TXFRAMEEND

Address offset: 0x190

Publish configuration for event TXFRAMEEND

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXFRAMEEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.35 PUBLISH_RXFRAMESTART

Address offset: 0x194

Publish configuration for event RXFRAMESTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXFRAMESTART will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.36 PUBLISH_RXFRAMEEND

Address offset: 0x198

Publish configuration for event RXFRAMEEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXFRAMEEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.37 PUBLISH_ERROR

Address offset: 0x19C

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.38 PUBLISH_RXERROR

Address offset: 0x1A8

Publish configuration for event RXERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.39 PUBLISH_ENDRX

Address offset: 0x1AC

Publish configuration for event ENDRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.40 PUBLISH_ENDTX

Address offset: 0x1B0

Publish configuration for event ENDTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.41 PUBLISH_AUTOCOLRESSTARTED

Address offset: 0x1B8

Publish configuration for event AUTOCOLRESSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event AUTOCOLRESSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.42 PUBLISH_COLLISION

Address offset: 0x1C8

Publish configuration for event COLLISION

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event COLLISION will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.43 PUBLISH_SELECTED

Address offset: 0x1CC

Publish configuration for event SELECTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SELECTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.44 PUBLISH_STARTED

Address offset: 0x1D0

Publish configuration for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.20.14.45 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

ID	R/W	Field	Value ID	Value	Description
A	RW	FIELDDETECTED_ACTIVATE			Shortcut between event FIELDDETECTED and task ACTIVATE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	FIELDLOST_SENSE			Shortcut between event FIELDLOST and task SENSE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	TXFRAMEEND_ENABLERXDATA			Shortcut between event TXFRAMEEND and task ENABLERXDATA
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.20.14.46 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Enable or disable interrupt for event READY
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	FIELDDETECTED			Enable or disable interrupt for event FIELDDETECTED
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	FIELDLOST			Enable or disable interrupt for event FIELDLOST
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	TXFRAMESTART			Enable or disable interrupt for event TXFRAMESTART
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXFRAMEEND			Enable or disable interrupt for event TXFRAMEEND
			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				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Enable
F	RW	RXFRAMESTART			Enable or disable interrupt for event RXFRAMESTART
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXFRAMEEND			Enable or disable interrupt for event RXFRAMEEND
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	RXERROR			Enable or disable interrupt for event RXERROR
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	ENDRX			Enable or disable interrupt for event ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	ENDTX			Enable or disable interrupt for event ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	AUTOCOLRESSTARTED			Enable or disable interrupt for event AUTOCOLRESSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	COLLISION			Enable or disable interrupt for event COLLISION
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	SELECTED			Enable or disable interrupt for event SELECTED
			Disabled	0	Disable
			Enabled	1	Enable
T	RW	STARTED			Enable or disable interrupt for event STARTED
			Disabled	0	Disable
			Enabled	1	Enable

7.20.14.47 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for event READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FIELDDETECTED			Write '1' to enable interrupt for event FIELDDETECTED
			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				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
C	RW	FIELDLOST			Write '1' to enable interrupt for event FIELDLOST
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXFRAMESTART			Write '1' to enable interrupt for event TXFRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXFRAMEEND			Write '1' to enable interrupt for event TXFRAMEEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXFRAMESTART			Write '1' to enable interrupt for event RXFRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXFRAMEEND			Write '1' to enable interrupt for event RXFRAMEEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	RXERROR			Write '1' to enable interrupt for event RXERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDRX			Write '1' to enable interrupt for event ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDTX			Write '1' to enable interrupt for event ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	AUTOCOLRESSTARTED			Write '1' to enable interrupt for event AUTOCOLRESSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	COLLISION			Write '1' to enable interrupt for event COLLISION
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	SELECTED			Write '1' to enable interrupt for event SELECTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	STARTED			Write '1' to enable interrupt for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.20.14.48 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FIELDDETECTED			Write '1' to disable interrupt for event FIELDDETECTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	FIELDLOST			Write '1' to disable interrupt for event FIELDLOST
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	TXFRAMESTART			Write '1' to disable interrupt for event TXFRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXFRAMEEND			Write '1' to disable interrupt for event TXFRAMEEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	RXFRAMESTART			Write '1' to disable interrupt for event RXFRAMESTART
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXFRAMEEND			Write '1' to disable interrupt for event RXFRAMEEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	RXERROR			Write '1' to disable interrupt for event RXERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				T S R N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
L	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	ENDTX			Write '1' to disable interrupt for event ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	AUTOCOLRESSTARTED			Write '1' to disable interrupt for event AUTOCOLRESSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	COLLISION			Write '1' to disable interrupt for event COLLISION
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	SELECTED			Write '1' to disable interrupt for event SELECTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.20.14.49 ERRORSTATUS

Address offset: 0x404

NFC Error Status register

Note: Write a bit to 1 to clear it. Writing 0 has no effect.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	FRAMEDELAYTIMEOUT			No STARTTX task triggered before expiration of the time set in FRAMEDELAYMAX

7.20.14.50 FRAMESTATUS.RX

Address offset: 0x40C

Result of last incoming frame

Note: Write a bit to 1 to clear it. Writing 0 has no effect.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CRCERROR			No valid end of frame (EoF) 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

7.20.14.51 NFCTAGSTATE

Address offset: 0x410

Current operating state of NFC tag

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

ID	R/W	Field	Value ID	Value	Description
A	R	NFCTAGSTATE			NfcTag state
			Disabled	0	Disabled or sense
			RampUp	2	RampUp
			Idle	3	Idle
			Receive	4	Receive
			FrameDelay	5	FrameDelay
			Transmit	6	Transmit

7.20.14.52 SLEEPSTATE

Address offset: 0x420

Sleep state during automatic collision resolution

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

ID	R/W	Field	Value ID	Value	Description
A	R	SLEEPSTATE			Reflects the sleep state during automatic collision resolution. Set to IDLE by a GOIDLE task. Set to SLEEP_A when a valid SLEEP_REQ frame is received or by a GOSLEEP task.
			Idle	0	State is IDLE.
			SleepA	1	State is SLEEP_A.

7.20.14.53 FIELDPRESENT

Address offset: 0x43C

Indicates the presence or not of a valid field

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

ID	R/W	Field	Value ID	Value	Description
A	R	FIELDPRESENT			Indicates if a valid field is present. Available only in the activated state.
			NoField	0	No valid field detected
			FieldPresent	1	Valid field detected
B	R	LOCKDETECT			Indicates if the low level has locked to the field
			NotLocked	0	Not locked to field
			Locked	1	Locked to field

7.20.14.54 FRAMEDELAYMIN

Address offset: 0x504

Minimum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000480				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMIN			Minimum frame delay in number of 13.56 MHz clock cycles

7.20.14.55 FRAMEDELAYMAX

Address offset: 0x508

Maximum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00001000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMAX			Maximum frame delay in number of 13.56 MHz clock cycles

7.20.14.56 FRAMEDELAYMODE

Address offset: 0x50C

Configuration register for the Frame Delay Timer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	FRAMEDELAYMODE			Configuration register for the Frame Delay Timer
			FreeRun	0	Transmission is independent of frame timer and will start when the STARTTX task is triggered. No timeout.
			Window	1	Frame is transmitted between FRAMEDELAYMIN and FRAMEDELAYMAX
			ExactVal	2	Frame is transmitted exactly at FRAMEDELAYMAX
			WindowGrid	3	Frame is transmitted on a bit grid between FRAMEDELAYMIN and FRAMEDELAYMAX

7.20.14.57 PACKETPTR

Address offset: 0x510

Packet pointer for TXD and RXD data storage in Data RAM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Packet pointer for TXD and RXD data storage in Data RAM. This address is a byte-aligned RAM address. **Note:** See the memory chapter for details about which memories are available for EasyDMA.

7.20.14.58 MAXLEN

Address offset: 0x514

Size of the RAM buffer allocated to TXD and RXD data storage each

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	MAXLEN		[0..257]	Size of the RAM buffer allocated to TXD and RXD data storage each

7.20.14.59 TXD.FRAMECONFIG

Address offset: 0x518

Configuration of outgoing frames

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000017				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PARITY			Indicates if parity is added to the frame
			NoParity	0	Parity is not added to TX frames
			Parity	1	Parity is added to TX frames
B	RW	DISCARDMODE			Discarding unused bits at start or end of a frame
			DiscardEnd	0	Unused bits are discarded at end of frame (EoF)
			DiscardStart	1	Unused bits are discarded at start of frame (SoF)
C	RW	SOF			Adding SoF or not in TX frames
			NoSoF	0	SoF symbol not added
			SoF	1	SoF symbol added
D	RW	CRCMODETX			CRC mode for outgoing frames
			NoCRCTX	0	CRC is not added to the frame
			CRC16TX	1	16 bit CRC added to the frame based on all the data read from RAM that is used in the frame

7.20.14.60 TXD.AMOUNT

Address offset: 0x51C

Size of outgoing frame

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TXDATABITS		[0..7]	Number of bits in the last or first byte read from RAM that shall be included in the frame (excluding parity bit). The DISCARDMODE field in FRAMECONFIG.TX selects if unused bits is discarded at the start or at the end of a frame. A value of 0 data bytes and 0 data bits is invalid.
B	RW	TXDATABYTES		[0..257]	Number of complete bytes that shall be included in the frame, excluding CRC, parity, and framing.

7.20.14.61 RXD.FRAMECONFIG

Address offset: 0x520

Configuration of incoming frames

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000015				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PARITY			Indicates if parity expected in RX frame
			NoParity	0	Parity is not expected in RX frames
			Parity	1	Parity is expected in RX frames
B	RW	SOF			SoF expected or not in RX frames
			NoSoF	0	SoF symbol is not expected in RX frames
			SoF	1	SoF symbol is expected in RX frames
C	RW	CRCMODERX			CRC mode for incoming frames
			NoCRCRX	0	CRC is not expected in RX frames
			CRC16RX	1	Last 16 bits in RX frame is CRC, CRC is checked and CRCSTATUS updated

7.20.14.62 RXD.AMOUNT

Address offset: 0x524

Size of last incoming frame

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

7.20.14.63 MODULATIONCTRL

Address offset: 0x52C

Enables the modulation output to a GPIO pin which can be connected to a second external antenna.

See MODULATIONPSEL for GPIO 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 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	MODULATIONCTRL			Configuration of modulation control.
			Invalid	0x0	Invalid, defaults to same behaviour as for Internal
			Internal	0x1	Use internal modulator only
			ModToGpio	0x2	Output digital modulation signal to a GPIO pin.
			InternalAndModToGpio	0x3	Use internal modulator and output digital modulation signal to a GPIO pin.

7.20.14.64 MODULATIONPSEL

Address offset: 0x538

Pin select for Modulation control

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

7.20.14.65 NFCID1_LAST

Address offset: 0x590

Last NFCID1 part (4, 7 or 10 bytes ID)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00006363				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1**
ID	R/W	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

7.20.14.66 NFCID1_2ND_LAST

Address offset: 0x594

Second last NFCID1 part (7 or 10 bytes ID)

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

7.20.14.67 NFCID1_3RD_LAST

Address offset: 0x598

Third last NFCID1 part (10 bytes ID)

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

7.20.14.68 AUTOCOLRESCONFIG

Address offset: 0x59C

Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is activated.

Note: When modifiying this register, bit `1` must be written to `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
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MODE			Enables/disables auto collision resolution
			Enabled	0	Auto collision resolution enabled
			Disabled	1	Auto collision resolution disabled

7.20.14.69 SENSRES

Address offset: 0x5A0

NFC-A SENS_RES auto-response settings

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

ID	R/W	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.

7.20.14.70 SELRES

Address offset: 0x5A4

NFC-A SEL_RES auto-response settings

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

ID	R/W	Field	Value ID	Value	Description
A	RW	RFU10			Reserved for future use. Shall be 0.
B	RW	CASCADE			Cascade as defined by the b3 of SEL_RES response in the NFC Forum, NFC Digital Protocol Technical Specification (controlled by hardware, shall be 0)
C	RW	RFU43			Reserved for future use. Shall be 0.
D	RW	PROTOCOL			Protocol as defined by the b7:b6 of SEL_RES response in the NFC Forum, NFC Digital Protocol Technical Specification
E	RW	RFU7			Reserved for future use. Shall be 0.

7.20.15 Electrical specification

7.20.15.1 NFCT Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
f_c	Frequency of operation		13.56		MHz
C_{MI}	Carrier modulation index	95			%
DR	Data Rate		106		kbps
V_{sense}	Peak differential field detect threshold level on NFC1-NFC2, with input being high impedance in sense mode		1.2		Vp
I_{max}	Maximum input current on NFCT pins			80	mA

7.20.15.2 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, excluding voltage supply and oscillator startup times			500	µs
t_{sense}	Time from remote field is present in SENSE mode to FIELDDETECTED event is asserted			20	µs

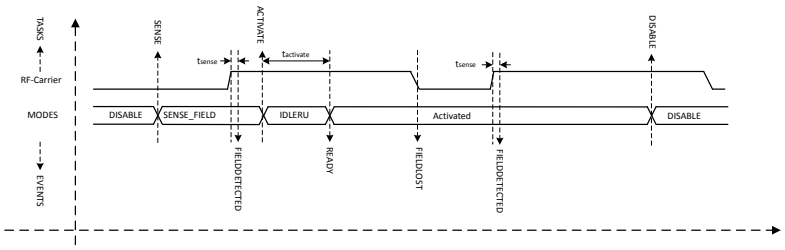

Figure 98: NFCT timing parameters (Shortcuts for FIELDDETECTED and FIELDLOST are disabled)

7.21 NVMC — Non-volatile memory controller

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

The NVMC is a split security peripheral. This means that when the NVMC is configured as non-secure, only a subset of the registers is available from the non-secure code. See SPU — System protection unit on page 584 and Registers on page 332 for more details.

When the NVMC is configured to be a secure peripheral, only secure code has access.

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 333. 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.

7.21.1 Writing to flash

When writing is enabled, in CONFIG register for secure region, or in CONFIGNS register for non-secure region, flash is written by writing a full 32-bit word to a word-aligned address in flash.

Secure code has access to both secure and non-secure regions, by using the appropriate configuration of CONFIG and CONFIGNS registers. Non-secure code, in constrast, has access to non-secure regions only. Thus, non-secure code only needs CONFIGNS.

The NVMC is only able to write `0` to erased bits in flash, that is bits set to `1`. It cannot write a bit back to `1`.

As illustrated in Memory on page 18, flash is divided into multiple pages. The same address in flash can only be written n_{WRITE} number of times before a page erase must be performed.

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`. The restriction about the number of writes (see above) still applies in this case.

The time it takes to write a word to flash is specified by t_{WRITE}. If CPU executes code from flash while the NVMC is writing to flash, the CPU will be stalled.

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

7.21.2 Erasing a secure page in flash

When secure region erase is enabled (in CONFIG register), a flash page can be erased by writing `0xFFFFFFFF` into the first 32-bit word in a flash page.

Page erase is only applicable to the code area in the flash and does not work with UICR.

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 stalled if the CPU executes code from the flash while the NVMC performs the erase operation.

See Partial erase of a page in flash for information on splitting the erase time in smaller chunks.

7.21.3 Erasing a non-secure page in flash

When non-secure region erase is enabled, a non-secure flash page can be erased by writing `0xFFFFFFFF` into the first 32-bit word of the flash page.

Page erase is only applicable to the code area in the flash and does not work with UICR.

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 stalled if the CPU executes code from the flash while the NVMC performs the erase operation.

7.21.4 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 is only accessible by secure code. Any write from non-secure code will be faulted.

To lock the chip after uploading non-secure code, do the fololwing steps:

1. Block access to secure code by setting UICR register to protected.
2. Use the register WRITEUICRNS on page 335, via non-secure debugger, to set APPROTECT (APPROTECT is automatically written to `0x00000000` by the NVMC).

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

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

7.21.5 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).

This functionality can be blocked by some configuration of the UICR protection bits, see the table NVMC protection on page 331.

The time it takes to perform an ERASEALL on page 333 command is specified by $t_{ERASEALL}$. The CPU is stalled if the CPU executes code from the flash while the NVMC performs the erase operation.

7.21.6 NVMC protection mechanisms

This chapter describes the different protection mechanisms for the non-volatile memory.

7.21.6.1 NVMC blocking

NVM integrity is assured through use of multiple levels of protection. Protection mechanisms can be configured to allow or block certain operations.

The table below shows the different status of protection bits, and which operations are allowed or blocked.

Protection bit status			NVMC protection	
SECURE APPROTECT	**APPROTECT**	**ERASE PROTECT**	**CTRL-AP ERASEALL**	**NVMC ERASEALL**
0	0	0	Available	Available
1	X	0	Available	Blocked
X	1	0	Available	Blocked
X	X	1	Blocked	Blocked
1 - Enabled, 0 - Disabled, X - Don't care				

Table 102: NVMC protection

Uploading code with secure debugging blocked

Non-secure code can program non-secure flash regions. To perform these operations, the NVMC has the following non-secure registers: CONFIGNS, READY, and READYNEXT.

CONFIGNS on page 335 works as the CONFIG register but it is used only for non-secure transactions. Both page erase and writing inside the flash require a write transaction (see Erasing a secure page in flash on page 330 or Erasing a non-secure page in flash on page 330). Because of this, the SPU — System protection unit on page 584 will guarantee that the non-secure code cannot write inside a secure page, since the transaction will never reach the NVMC controller.

7.21.6.2 NVMC power failure protection

NVMC power failure protection is possible through the use of a power-fail comparator that is monitoring power supply.

If the power-fail comparator is enabled, and the power supply voltage is below V_{POF} threshold, the power-fail comparator will prevent the NVMC from performing erase or write operations in non-volatile memory (NVM).

If a power failure warning is present at the start of an NVM write or erase operation, the NVMC will block the operation and a bus error will be signalled. If a power failure warning occurs during an ongoing NVM write operation, the NVMC will try to finish the operation. And if the power failure warning persists, consecutive NVM write operations will be blocked by the NVMC, and a bus error will be signalled.

7.21.7 Cache

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

See Memory on page 18 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, see CPU parameter W_FLASHCACHE.

Enabling the cache can increase the 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 draws 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 be reduced.

When disabled, the cache does not draw current and its content is not retained.

It is possible to enable cache profiling to analyze the performance of the cache for your program using the register ICACHECNF. When profiling is enabled, registers IHIT and IMISS are incremented for every instruction cache hit or miss respectively.

7.21.8 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50039000 0x40039000	APPLICATION	NVMC	NVMC : S NVMC : NS	SPLIT	NA	Non-volatile memory controller	NVMC built-in cache not supported (ICACHECNF, IHIT and IMISS registers).
0x41080000	NETWORK	NVMC	NVMC	NS	NA	Non-Volatile Memory Controller	NVMC TrustZone registers (WRITEUICRNS and CONFIGNS) not supported.

Table 103: Instances

Register	Offset	Security	Description
READY	0x400	NS	Ready flag
READYNEXT	0x408	NS	Ready flag
CONFIG	0x504	S	Configuration register
ERASEALL	0x50C	S	Register for erasing all non-volatile user memory
ERASEPAGEPARTIALCFG	0x51C	S	Register for partial erase configuration
ICACHECNF	0x540	S	I-code cache configuration register
IHIT	0x548	S	I-code cache hit counter
IMISS	0x54C	S	I-code cache miss counter
CONFIGNS	0x584	NS	Non-secure configuration register
WRITEUICRNS	0x588	NS	Non-secure APPROTECT enable register

Table 104: Register overview

7.21.8.1 READY

Address offset: 0x400

Ready flag

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

7.21.8.2 READYNEXT

Address offset: 0x408

Ready flag

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	R	READYNEXT			NVMC can accept a new write operation
			Busy	0	NVMC cannot accept any write operation
			Ready	1	NVMC is ready

7.21.8.3 CONFIG

Address offset: 0x504

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	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. Using other values than those defined may result in unpredictable behavior.
			Ren	0	Read only access
			Wen	1	Write enabled
			Een	2	Erase enabled
			PEen	4	Partial erase enabled

7.21.8.4 ERASEALL

Address offset: 0x50C

Register for erasing all non-volatile user memory

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

7.21.8.5 ERASEPAGEPARTIALCFG

Address offset: 0x51C

Register for partial erase configuration

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

7.21.8.6 ICACHECNF

Address offset: 0x540

I-code cache configuration register

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

7.21.8.7 IHIT

Address offset: 0x548

I-code cache hit counter

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

7.21.8.8 IMISS

Address offset: 0x54C

I-code cache miss counter

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

7.21.8.9 CONFIGNS

Address offset: 0x584

Non-secure configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	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. Using other values than those defined may result in unpredictable behavior.
			Ren	0	Read only access
			Wen	1	Write enabled
			Een	2	Erase enabled

7.21.8.10 WRITEUICRNS

Address offset: 0x588

Non-secure APPROTECT 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				B B B B B B B B B B B B B B B B B B B B B B B B B B B B B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	SET			Allow non-secure code to set APPROTECT
			Set	1	Set value
B	W	KEY			Key to write in order to validate the write operation
			Keyvalid	0xAFBE5A7	Key value

7.21.9 Electrical specification

7.21.9.1 Flash programming

Symbol	Description	Min.	Typ.	Max.	Units
n_{WRITE}	Number of times a 32-bit word can be written before erase			2	
$n_{ENDURANCE}$	Erase cycles per page	10000			
t_{WRITE}	Time to write one 32-bit word			43[10]	µs
$t_{ERASEPAGE}$	Time to erase one page			87.5[10]	ms
$t_{ERASEALL}$	Time to erase all flash			173[10]	ms
$t_{ERASEPAGEPARTIAL,setu}$	Setup time for one partial erase	..	..	..	ms

7.21.9.2 Cache size

Symbol	Description	Min.	Typ.	Max.	Units
$Size_{ICODE}$	I-Code cache size		2048[11]		Bytes

7.22 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
- Selectable ratio of 64 or 80 between PDM_CLK and output sample rate

The PDM module illustrated below is interfacing up to two digital microphones with the PDM interface. EasyDMA is implemented to relieve the real-time requirements associated with controlling of the PDM slave from a low priority CPU execution context. It also includes all the necessary digital filter elements to produce pulse code modulation (PCM) samples. The PDM module allows continuous audio streaming.

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

[11] Only applicable for network core. Application core has a separate cache, see CACHE — Instruction/data cache on page 108.

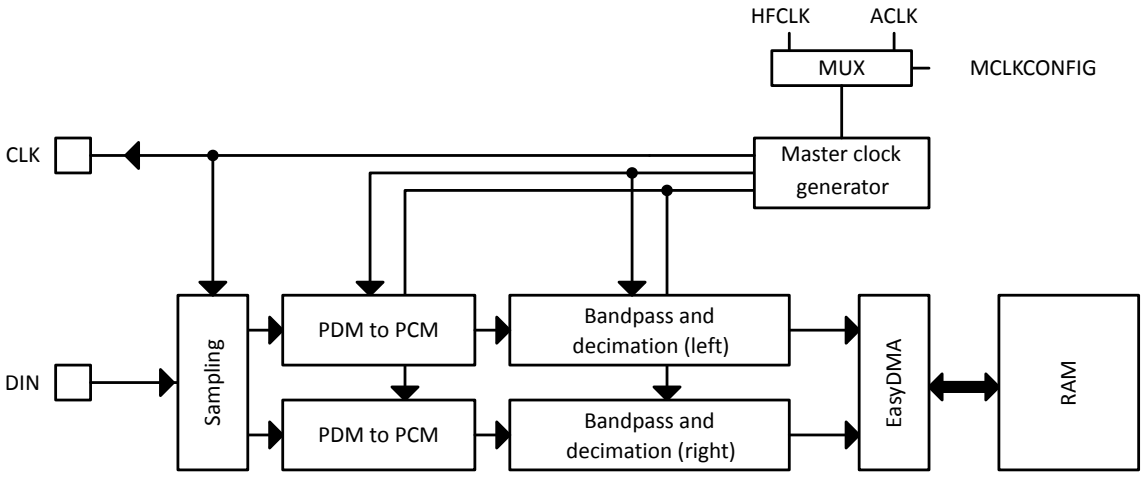

Figure 99: PDM module

7.22.1 Master clock source selection

The master clock source can be configured in register MCLKCONFIG on page 349. Choose one of the following as the master clock source:

- 32 MHz peripheral clock (PCLK32M), synchronous to HFCLK.
- Audio PLL clock (ACLK) with configurable frequency.

The peripheral must be stopped before selecting the master clock source. The use of the STOP task and the STOPPED event is described in Module operation on page 338.

To improve the master clock accuracy and jitter performance, it is recommended (but not mandatory) that the PCLK32M source is running off the HFXO instead of the HFINT oscillator. The ACLK source requires the use of HFXO. See CLOCK — Clock control on page 69 for more information about starting HFXO for the relevant clock source.

7.22.2 Master clock generator

The master clock generator'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.

The following equation can be used to calculate the value of the PDMCLKCTRL register for a given PDM clock- and master clock source frequency:

$$PDMCLKCTRL = 4096 \cdot \left\lfloor \frac{f_{pdm} \cdot 1048576}{f_{source} + \frac{f_{pdm}}{2}} \right\rfloor$$

Figure 100: PDM clock frequency equation

Where f_{pdm} is the requested PDM clock frequency in Hz, and f_{source} is the master clock generator source in Hz. Because of rounding errors, an accurate PDM clock may not be achievable. The actual PDM frequency can be calculated using the equation below.

$$f_{actual} = \frac{f_{source}}{\left\lfloor \frac{1048576 \cdot 4096}{PDMCLKCTRL} \right\rfloor}$$

Figure 101: Actual PDM frequency

The clock error can be calculated using the equation below. The error *e* is the percentage difference from the requested f_{pdm} frequency.

$$e = 100 \cdot \frac{f_{actual} - f_{pdm}}{f_{pdm}} = 100 \cdot \frac{\frac{f_{source}}{\left\lfloor \frac{1048576 \cdot 4096}{PDMCLKCTRL} \right\rfloor} - f_{pdm}}{f_{pdm}}$$

Figure 102: PDM frequency error equation

The PDM frequency can be adjusted while the clock generator is running.

- For PCLK32M, by using PDMCLKCTRL
- For ACLK, by adjusting the audio clock source, see CLOCK — Clock control on page 69.

Requested PDM frequency f_{pdm} [Hz]	f_{source} [Hz]	RATIO	PDMCLKCTRL	Actual PDM frequency f_{actual} [Hz]	Sample frequency [Hz]	Error [%]
1024000	32000000 (PCLK32M)	64	135274496	1032258.1	16129.0	0.81
1280000	32000000 (PCLK32M)	80	168427520	1280000	16000	0
1024000	12288000 (ACLK)	64	343597056	1024000	16000	0

Table 105: Configuration examples

7.22.3 Module operation

By default, bits from the left PDM microphone are sampled on PDM_CLK falling edge, and 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, then 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 is 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 behavior.

7.22.4 Decimation filter

In order to convert the incoming data stream into PCM audio samples, a decimation filter is included in the PDM interface module.

The input of the filter is the two-channel PDM serial stream (with left channel on clock high, right channel on clock low). Depending on the RATIO selected, its output is 2 × 16-bit PCM samples at a sample rate either 64 times or 80 times (depending on the RATIO register) lower than the PDM clock rate.

The filter stage of each channel is followed by a digital volume control, to attenuate or amplify the output samples in a range of -20 dB to +20 dB around the default (reset) setting, defined by $G_{PDM,default}$. The gain is controlled by the GAINL and GAINR registers.

As an example, if the goal is to achieve 2500 RMS output samples (16-bit) with a 1 kHz 90 dBA signal into a -26 dBFS sensitivity PDM microphone, do the following:

- Sum the PDM module's default gain ($G_{PDM,default}$) and the gain introduced by the microphone and acoustic path of his implementation (an attenuation would translate into a negative gain)
- Adjust GAINL and GAINR by the above summed amount. Assuming that only the PDM module influences the gain, GAINL and GAINR must be set to $-G_{PDM,default}$ dB to achieve the requirement.

With $G_{PDM,default}$ = 3.2 dB, and as GAINL and GAINR are expressed in 0.5 dB steps, the closest value to program would be 3.0 dB, which can be calculated as:

```
GAINL = GAINR = (DefaultGain - (2 * 3))
```

Remember to check that the resulting values programmed into GAINL and GAINR fall within MinGain and MaxGain.

7.22.5 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 18 for more information about the different memory regions.

The DMA transfer supports Stereo (left and right 16-bit samples) and Mono (left only) data transfer as configured in the OPERATION field of the MODE register. The samples are stored little endian.

MODE.OPERATION	Bits per sample	Result stored per RAM word	Physical RAM allocated (32-bit words)	Result boundary indexes in RAM	Note
Stereo	32 (2x16)	L+R	ceil(SAMPLE.MAXCNT/2)	R0=[31:16]; L0=[15:0]	Default
Mono	16	2xL	ceil(SAMPLE.MAXCNT/2)	L1=[31:16]; L0=[15:0]	

Table 106: DMA sample storage

The destination buffer in RAM consists of one block, the size of which is set in SAMPLE.MAXCNT register. Format is number of 16-bit samples. The physical RAM allocated is always:

```
(RAM allocation, in bytes) = SAMPLE.MAXCNT * 2;
```

(but the mapping of the samples depends on MODE.OPERATION.

If OPERATION=Stereo, RAM will contain a succession of left and right samples.

If OPERATION=Mono, RAM will contain a succession of left only 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.

7.22.6 Hardware example

PDM can be configured with a single microphone (mono), or with two microphones.

When a single microphone is used, connect the microphone clock to CLK, and data to DIN.

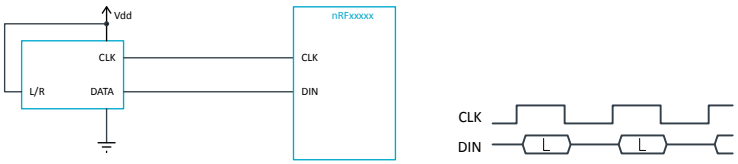

Figure 103: Example of a single PDM microphone, wired as left

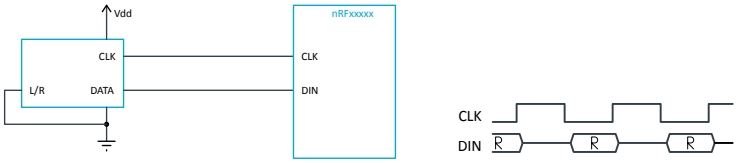

Figure 104: Example of a single PDM microphone, wired as right

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.

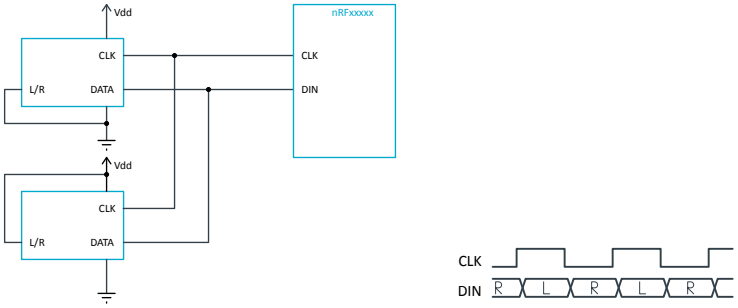

Figure 105: Example of two PDM microphones

7.22.7 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 control on page 43 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 behavior in the PDM module, the pins used by the PDM module must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 341

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 behavior.

PDM signal	PDM pin	Direction	Output value	Comment
CLK	As specified in PSEL.CLK	Output	0	
DIN	As specified in PSEL.DIN	Input	Not applicable	

Table 107: GPIO configuration before enabling peripheral

7.22.8 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50026000 0x40026000	APPLICATION	PDM	PDM0 : S PDM0 : NS	US	SA	Pulse density modulation (digital microphone) interface	

Table 108: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Starts continuous PDM transfer
TASKS_STOP	0x004		Stops PDM transfer
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
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
PUBLISH_STARTED	0x180		Publish configuration for event STARTED
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_END	0x188		Publish configuration for event END
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		PDM module enable register
PDMCLKCTRL	0x504		PDM clock generator control
MODE	0x508		Defines the routing of the connected PDM microphones' signals
GAINL	0x518		Left output gain adjustment
GAINR	0x51C		Right output gain adjustment
RATIO	0x520		Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.
PSEL.CLK	0x540		Pin number configuration for PDM CLK signal
PSEL.DIN	0x544		Pin number configuration for PDM DIN signal
MCLKCONFIG	0x54C		Master clock generator configuration
SAMPLE.PTR	0x560		RAM address pointer to write samples to with EasyDMA
SAMPLE.MAXCNT	0x564		Number of samples to allocate memory for in EasyDMA mode

Table 109: Register overview

7.22.8.1 TASKS_START

Address offset: 0x000

Starts continuous PDM 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Starts continuous PDM transfer
			Trigger	1	Trigger task

7.22.8.2 TASKS_STOP

Address offset: 0x004

Stops PDM 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stops PDM transfer
			Trigger	1	Trigger task

7.22.8.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.22.8.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.22.8.5 EVENTS_STARTED

Address offset: 0x100

PDM transfer has started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			PDM transfer has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.22.8.6 EVENTS_STOPPED

Address offset: 0x104

PDM transfer has finished

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			PDM transfer has finished
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.22.8.7 EVENTS_END

Address offset: 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

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			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
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.22.8.8 PUBLISH_STARTED

Address offset: 0x180

Publish configuration for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.22.8.9 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.22.8.10 PUBLISH_END

Address offset: 0x188

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.22.8.11 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Enable or disable interrupt for event STARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	STOPPED			Enable or disable interrupt for event STOPPED
			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				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	R/W	Field	Value ID	Value	Description
C	RW	END			Enable or disable interrupt for event END
			Disabled	0	Disable
			Enabled	1	Enable

7.22.8.12 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.22.8.13 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.22.8.14 ENABLE

Address offset: 0x500

PDM module enable register

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable PDM module
			Disabled	0	Disable
			Enabled	1	Enable

7.22.8.15 PDMCLKCTRL

Address offset: 0x504

PDM clock generator control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x08400000				**0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	FREQ			PDM_CLK frequency configuration. Enumerations are deprecated, use PDMCLKCTRL equation to find the register value. The 12 least significant bits of the register are ignored and shall be set to zero.
			1000K	0x08000000	PDM_CLK = 32 MHz / 32 = 1.000 MHz
			Default	0x08400000	PDM_CLK = 32 MHz / 31 = 1.032 MHz. Nominal clock for RATIO=Ratio64.
			1067K	0x08800000	PDM_CLK = 32 MHz / 30 = 1.067 MHz
			1231K	0x09800000	PDM_CLK = 32 MHz / 26 = 1.231 MHz
			1280K	0x0A000000	PDM_CLK = 32 MHz / 25 = 1.280 MHz. Nominal clock for RATIO=Ratio80.
			1333K	0x0A800000	PDM_CLK = 32 MHz / 24 = 1.333 MHz

7.22.8.16 MODE

Address offset: 0x508

Defines the routing of the connected PDM microphones' signals

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	OPERATION			Mono or stereo operation
			Stereo	0	Sample and store one pair (left + right) of 16-bit samples per RAM word R=[31:16]; L=[15:0]
			Mono	1	Sample and store two successive left samples (16 bits each) per RAM word L1=[31:16]; L0=[15:0]
B	RW	EDGE			Defines on which PDM_CLK edge left (or mono) is sampled
			LeftFalling	0	Left (or mono) is sampled on falling edge of PDM_CLK
			LeftRising	1	Left (or mono) is sampled on rising edge of PDM_CLK

7.22.8.17 GAINL

Address offset: 0x518

Left output gain adjustment

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000028				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	GAINL			Left output gain adjustment, in 0.5 dB steps, around the default module gain (see electrical parameters)
					0x00 -20 dB gain adjust
					0x01 -19.5 dB gain adjust
					(...)
					0x27 -0.5 dB gain adjust
					0x28 0 dB gain adjust
					0x29 +0.5 dB gain adjust
					(...)
					0x4F +19.5 dB gain adjust
					0x50 +20 dB gain adjust
			MinGain	0x00	-20 dB gain adjustment (minimum)
			DefaultGain	0x28	0 dB gain adjustment
			MaxGain	0x50	+20 dB gain adjustment (maximum)

7.22.8.18 GAINR

Address offset: 0x51C

Right output gain adjustment

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000028				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	GAINR			Right output gain adjustment, in 0.5 dB steps, around the default module gain (see electrical parameters)
			MinGain	0x00	-20 dB gain adjustment (minimum)
			DefaultGain	0x28	0 dB gain adjustment
			MaxGain	0x50	+20 dB gain adjustment (maximum)

7.22.8.19 RATIO

Address offset: 0x520

Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RATIO			Selects the ratio between PDM_CLK and output sample rate
			Ratio64	0	Ratio of 64
			Ratio80	1	Ratio of 80

7.22.8.20 PSEL.CLK

Address offset: 0x540

Pin number configuration for PDM CLK signal

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

7.22.8.21 PSEL.DIN

Address offset: 0x544

Pin number configuration for PDM DIN signal

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

7.22.8.22 MCLKCONFIG

Address offset: 0x54C

Master clock generator 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	SRC			Master clock source selection
			PCLK32M	0	32 MHz peripheral clock
			ACLK	1	Audio PLL clock

7.22.8.23 SAMPLE.PTR

Address offset: 0x560

RAM address pointer to write samples to with EasyDMA

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SAMPLEPTR			Address to write PDM samples to over DMA

Note: See the memory chapter for details about which memories are available for EasyDMA.

7.22.8.24 SAMPLE.MAXCNT

Address offset: 0x564

Number of samples to allocate memory for in EasyDMA mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	BUFFSIZE		[0..32767]	Length of DMA RAM allocation in number of samples

7.22.9 Electrical specification

7.22.9.1 PDM Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$f_{PDM,CLK,64}$	PDM clock speed. PDMCLKCTRL = Default (Setting needed for 16 MHz sample frequency @ RATIO = Ratio64)		1.032		MHz
$f_{PDM,CLK,80}$	PDM clock speed. PDMCLKCTRL = 1280K (Setting needed for 16 MHz sample frequency @ RATIO = Ratio80)		1.280		MHz
$t_{PDM,JITTER}$	Jitter in PDM clock output			20	ns
$T_{dPDM,CLK}$	PDM clock duty cycle	40	50	60	%
$t_{PDM,DATA}$	Decimation filter delay			5	ms
$t_{PDM,cv}$	Allowed clock edge to data valid			125	ns
$t_{PDM,ci}$	Allowed (other) clock edge to data invalid	0			ns
$t_{PDM,s}$	Data setup time at $f_{PDM,CLK}$ = 1.024 MHz or 1.280 MHz	65			ns
$t_{PDM,h}$	Data hold time at $f_{PDM,CLK}$ = 1.024 MHz or 1.280 MHz	0			ns
$G_{PDM,default}$	Default (reset) absolute gain of the PDM module		3.2		dB

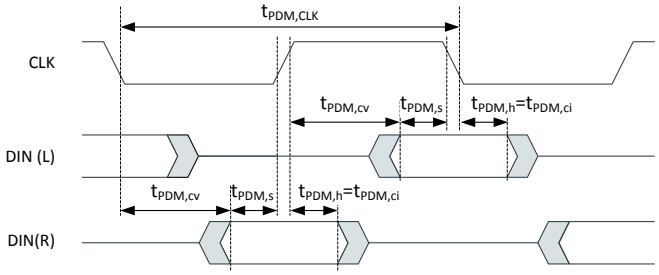

Figure 106: PDM timing diagram

7.23 PWM — Pulse width modulation

The pulse width modulation (PWM) module enables the generation of pulse width modulated signals on GPIO. The module implements an up or up-and-down counter with four PWM channels that drive assigned GPIOs.

The following are the main features of a PWM module:

- Programmable PWM frequency
- Up to four PWM channels with individual polarity and duty cycle values
- Edge or center-aligned pulses across PWM channels
- Multiple duty cycle arrays (sequences) defined in RAM
- Autonomous and glitch-free update of duty cycle values directly from memory through EasyDMA (no CPU involvement)
- Change of polarity, duty cycle, and base frequency possibly on every PWM period
- RAM sequences can be repeated or connected into loops

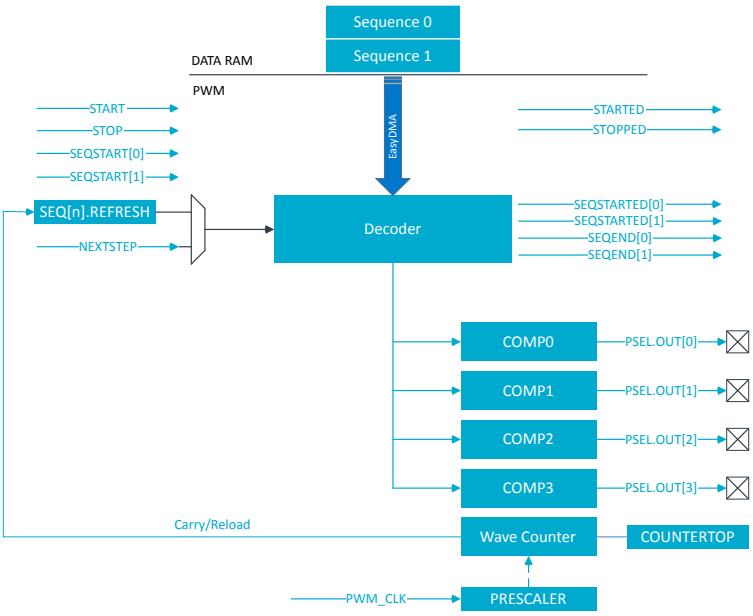

Figure 107: PWM module

7.23.1 Wave counter

The wave counter is responsible for generating the pulses at a duty cycle that depends on the compare values, and at a frequency that depends on COUNTERTOP.

There is one common 15-bit counter with four compare channels. Thus, all four channels will share the same period (PWM frequency), but can have individual duty cycle and polarity. The polarity is set by a value read from RAM (see figure Decoder memory access modes on page 354). Whether the counter counts up, or up and down, is controlled by the MODE register.

The timer top value is controlled by the COUNTERTOP register. This register value, in conjunction with the selected PRESCALER of the PWM_CLK, will result in a given PWM period. A COUNTERTOP value smaller than the compare setting will result in a state where no PWM edges are generated. OUT[n] is held high, given that the polarity is set to FallingEdge. All compare registers are internal and can only be configured through decoder presented later. COUNTERTOP can be safely written at any time.

Sampling follows the START task. If DECODER.LOAD=WaveForm, the register value is ignored and taken from RAM instead (see section Decoder with EasyDMA on page 354 for more details). If DECODER.LOAD is anything else than the WaveForm, it is sampled following a STARTSEQ[n] task and when loading a new value from RAM during a sequence playback.

The following figure shows the counter operating in up mode (MODE=PWM_MODE_Up), with three PWM channels with the same frequency but different duty cycle:

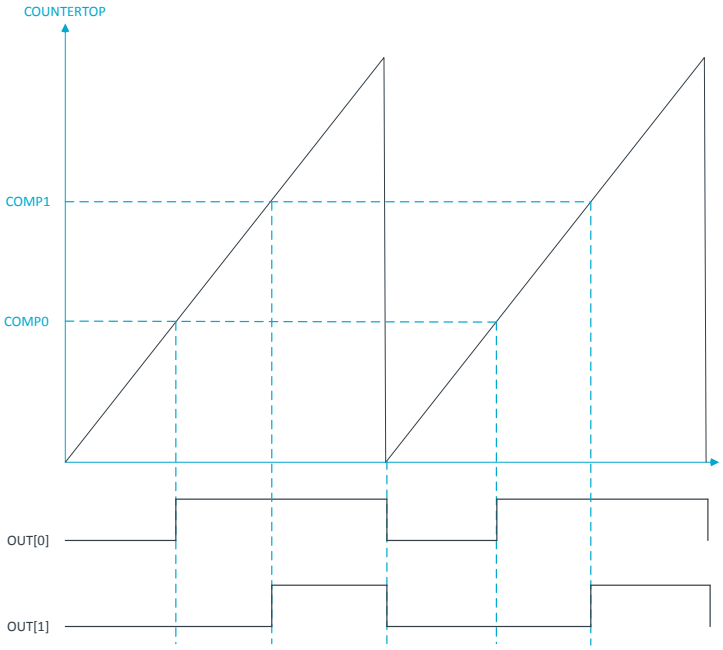

Figure 108: PWM counter in up mode example - FallingEdge polarity

The counter is automatically reset to zero when COUNTERTOP is reached and OUT[n] will invert. OUT[n] is held low if the compare value is 0 and held high if set to COUNTERTOP, given that the polarity is set to FallingEdge. Counter running in up mode results in pulse widths that are edge-aligned. The following is the code for the counter in up mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up mode, the following formula can be used to compute the PWM period and the step size:

PWM period: $T_{PWM(Up)} = T_{PWM\_CLK} * COUNTERTOP$

Step width/Resolution: $T_{steps} = T_{PWM\_CLK}$

The following figure shows the counter operating in up-and-down mode (MODE=PWM_MODE_UpAndDown), with two PWM channels with the same frequency but different duty cycle and output polarity:

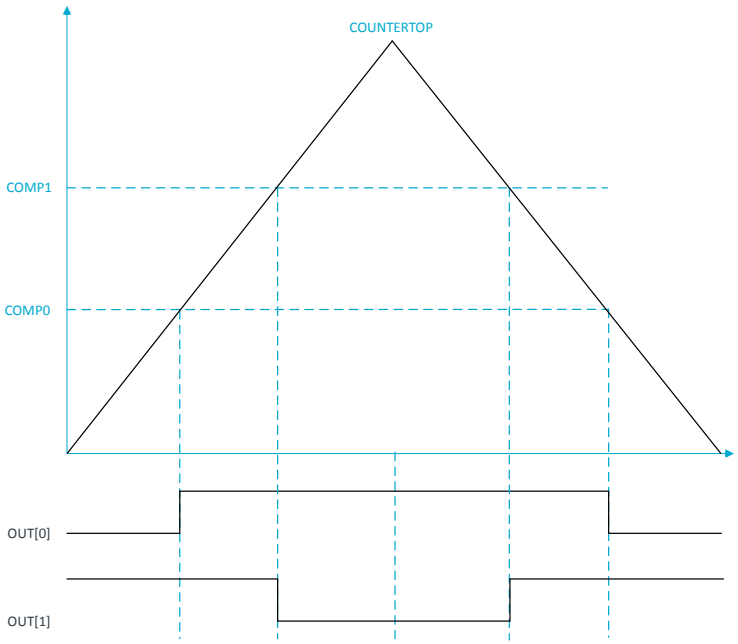

Figure 109: PWM counter in up-and-down mode example

The counter starts decrementing to zero when COUNTERTOP is reached and will invert the OUT[n] when compare value is hit for the second time. This results in a set of pulses that are center-aligned. The following is the code for the counter in up-and-down mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_UpAndDown << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up-and-down mode, the following formula can be used to compute the PWM period and the step size:

$T_{PWM(Up\ And\ Down)} = T_{PWM\_CLK} * 2 * COUNTERTOP$

Step width/Resolution: $T_{steps} = T_{PWM\_CLK} * 2$

7.23.2 Decoder with EasyDMA

The decoder uses EasyDMA to take PWM parameters stored in RAM and update the internal compare registers of the wave counter, based on the mode of operation.

PWM parameters are organized into a sequence containing at least one half word (16 bit). Its most significant bit[15] denotes the polarity of the OUT[n] while bit[14:0] is the 15-bit compare value.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Id				B A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
Id	RW	Field	Value Id	Value	Description
A	RW	COMPARE			Duty cycle setting - value loaded to internal compare register
B	RW	POLARITY			Edge polarity of GPIO.
			RisingEdge	0	First edge within the PWM period is rising
			FallingEdge	1	First edge within the PWM period is falling

The DECODER register controls how the RAM content is interpreted and loaded into the internal compare registers. The LOAD field controls if the RAM values are loaded to all compare channels, or to update a group or all channels with individual values. The following figure illustrates how parameters stored in RAM are organized and routed to various compare channels in different modes:

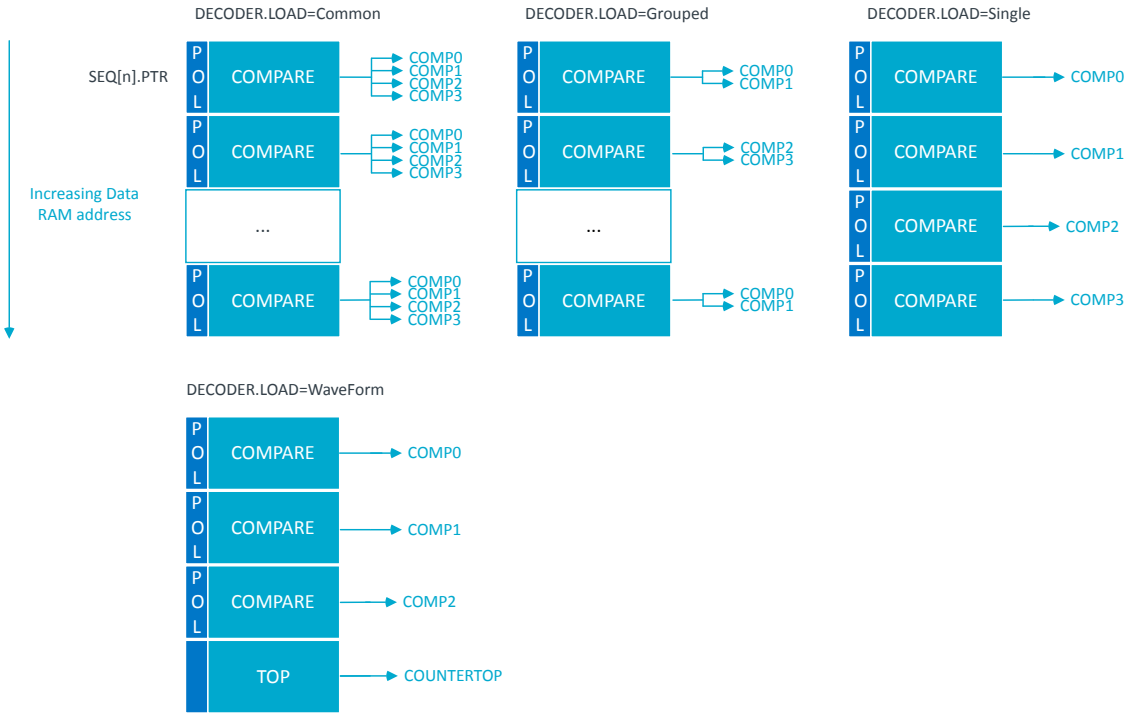

Figure 110: Decoder memory access modes

A special mode of operation is available when DECODER.LOAD is set to WaveForm. In this mode, up to three PWM channels can be enabled - OUT[0] to OUT[2]. In RAM, four values are loaded at a time: the first, second and third location are used to load the values, and the fourth RAM location is used to load

the COUNTERTOP register. This way one can have up to three PWM channels with a frequency base that changes on a per PWM period basis. This mode of operation is useful for arbitrary wave form generation in applications, such as LED lighting.

The register SEQ[n].REFRESH=N (one per sequence n=0 or 1) will instruct a new RAM stored pulse width value on every (N+1)th PWM period. Setting the register to zero will result in a new duty cycle update every PWM period, as long as the minimum PWM period is observed.

Note that registers SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored when DECODER.MODE=NextStep. The next value is loaded upon every received NEXTSTEP task.

SEQ[n].PTR is the pointer used to fetch COMPARE values from RAM. If the SEQ[n].PTR is not pointing to a RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 18 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 mode requires one half word per group, while the Single mode requires one half word per channel, thus increasing the RAM size occupation. If PWM generation is not running when the SEQSTART[n] task is triggered, the task will load the first value from RAM and then start the PWM generation. A SEQSTARTED[n] event is generated as soon as the EasyDMA has read the first PWM parameter from RAM and the wave counter has started executing it. When LOOP.CNT=0, sequence n=0 or 1 is played back once. After the last value in the sequence has been loaded and started executing, a SEQEND[n] event is generated. The PWM generation will then continue with the last loaded value. The following figure illustrates an example of a simple playback.

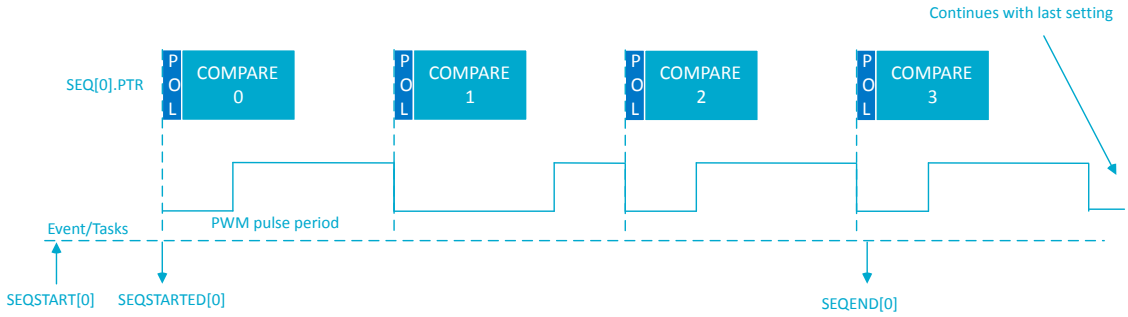

Figure 111: Simple sequence example

The following source code is used for configuration and timing details in a sequence where only sequence 0 is used and only run once with a new PWM duty cycle for each period.

```
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                                                PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

To completely stop the PWM generation and force the associated pins to a defined state, a STOP task can be triggered at any time. A STOPPED event is generated when the PWM generation has stopped at the end of the currently running PWM period, and the pins go into their idle state as defined in GPIO OUT register. PWM generation can then only be restarted through a SEQSTART[n] task. SEQSTART[n] will resume PWM generation after having loaded the first value from the RAM buffer defined in the SEQ[n].PTR register.

The following table indicates when specific registers get sampled by the hardware. Care should be taken when updating these registers to avoid that values are applied earlier than expected.

Register	Taken into account by hardware	Recommended (safe) update
SEQ[n].PTR	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[n].CNT	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[0].ENDDELAY	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [0] gets loaded from RAM (indicated by the SEQEND[0] event)
		At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].ENDDELAY	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [1] gets loaded from RAM (indicated by the SEQEND[1] event)
		At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
SEQ[0].REFRESH	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].REFRESH	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
COUNTERTOP	In DECODER.LOAD=WaveForm: this register is ignored.	Before starting PWM generation through a SEQSTART[n] task
	In all other LOAD modes: at the end of current PWM period (indicated by the PWMPERIODEND event)	After a STOP task has been triggered, and the STOPPED event has been received.
MODE	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
DECODER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PRESCALER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
LOOP	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PSEL.OUT[n]	Immediately	Before enabling the PWM instance through the ENABLE register

Table 110: When to safely update PWM registers

Note: SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored at the end of a complex sequence, indicated by a LOOPSDONE event. The reason for this is that the last value loaded from RAM is maintained until further action from software (restarting a new sequence, or stopping PWM generation).

The following figure shows a more complex example using the register LOOP on page 372.

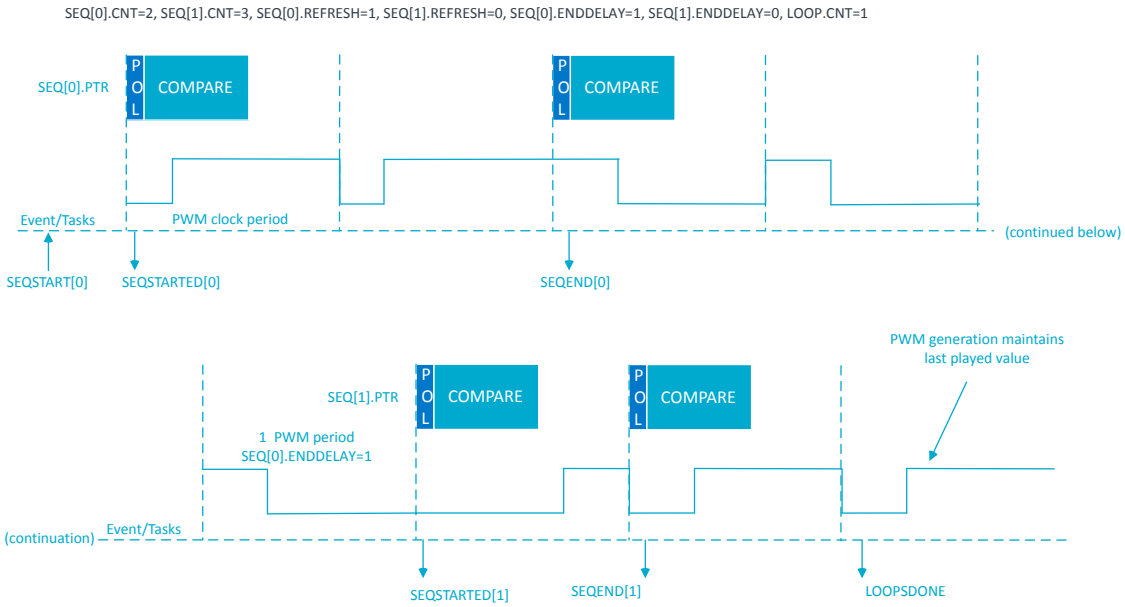

Figure 112: Example using two sequences

In this case, an automated playback takes place, consisting of SEQ[0], delay 0, SEQ[1], delay 1, then again SEQ[0], etc. The user can choose to start a complex playback with SEQ[0] or SEQ[1] through sending the SEQSTART[0] or SEQSTART[1] task. The complex playback always ends with delay 1.

The two sequences 0 and 1 are defined by the addresses of value tables in RAM (pointed to by SEQ[n].PTR) and the buffer size (SEQ[n].CNT). The rate at which a new value is loaded is defined individually for each sequence by SEQ[n].REFRESH. The chaining of sequence 1 following the sequence 0 is implicit, the LOOP.CNT register allows the chaining of sequence 1 to sequence 0 for a determined number of times. In other words, it allows to repeat a complex sequence a number of times in a fully automated way.

In the following code example, sequence 0 is defined with SEQ[0].REFRESH set to 1, meaning that a new PWM duty cycle is pushed every second PWM period. This complex sequence is started with the SEQSTART[0] task, so SEQ[0] is played first. Since SEQ[0].ENDDELAY=1 there will be one PWM period delay between last period on sequence 0 and the first period on sequence 1. Since SEQ[1].ENDDELAY=0 there is no delay 1, so SEQ[0] would be started immediately after the end of SEQ[1]. However, as LOOP.CNT is

1, the playback stops after having played SEQ[1] only once, and both SEQEND[1] and LOOPSDONE are generated (their order is not guaranteed in this case).

```
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                  PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                                                  PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (1 << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER   = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                      (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR  = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT  = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                                                  PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 1;
NRF_PWM0->SEQ[0].ENDDELAY = 1;
NRF_PWM0->SEQ[1].PTR  = ((uint32_t)(seq1_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[1].CNT  = ((sizeof(seq1_ram) / sizeof(uint16_t)) <<
                                                  PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[1].REFRESH  = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

The decoder can also be configured to asynchronously load new PWM duty cycle. If the DECODER.MODE register is set to NextStep, then the NEXTSTEP task will cause an update of internal compare registers on the next PWM period.

The following figures provide an overview of each part of an arbitrary sequence, in various modes (LOOP.CNT=0 and LOOP.CNT>0). In particular, the following are represented:

- Initial and final duty cycle on the PWM output(s)
- Chaining of SEQ[0] and SEQ[1] if LOOP.CNT>0
- Influence of registers on the sequence
- Events generated during a sequence
- DMA activity (loading of next value and applying it to the output(s))

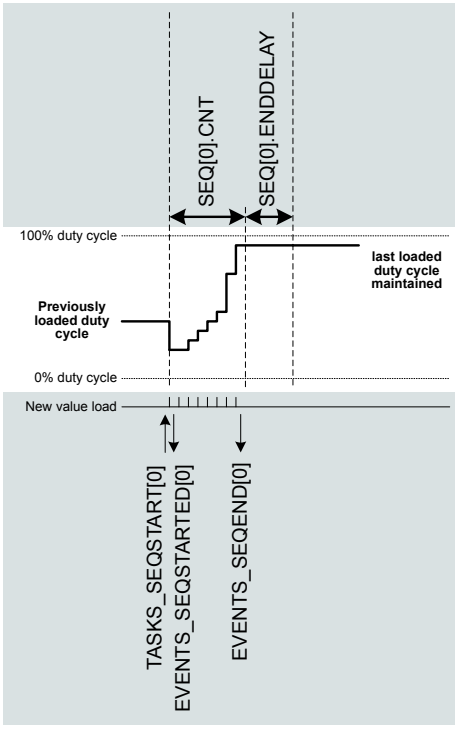

Figure 113: Single shot (LOOP.CNT=0)

Note: The single-shot example also applies to SEQ[1]. Only SEQ[0] is represented for simplicity.

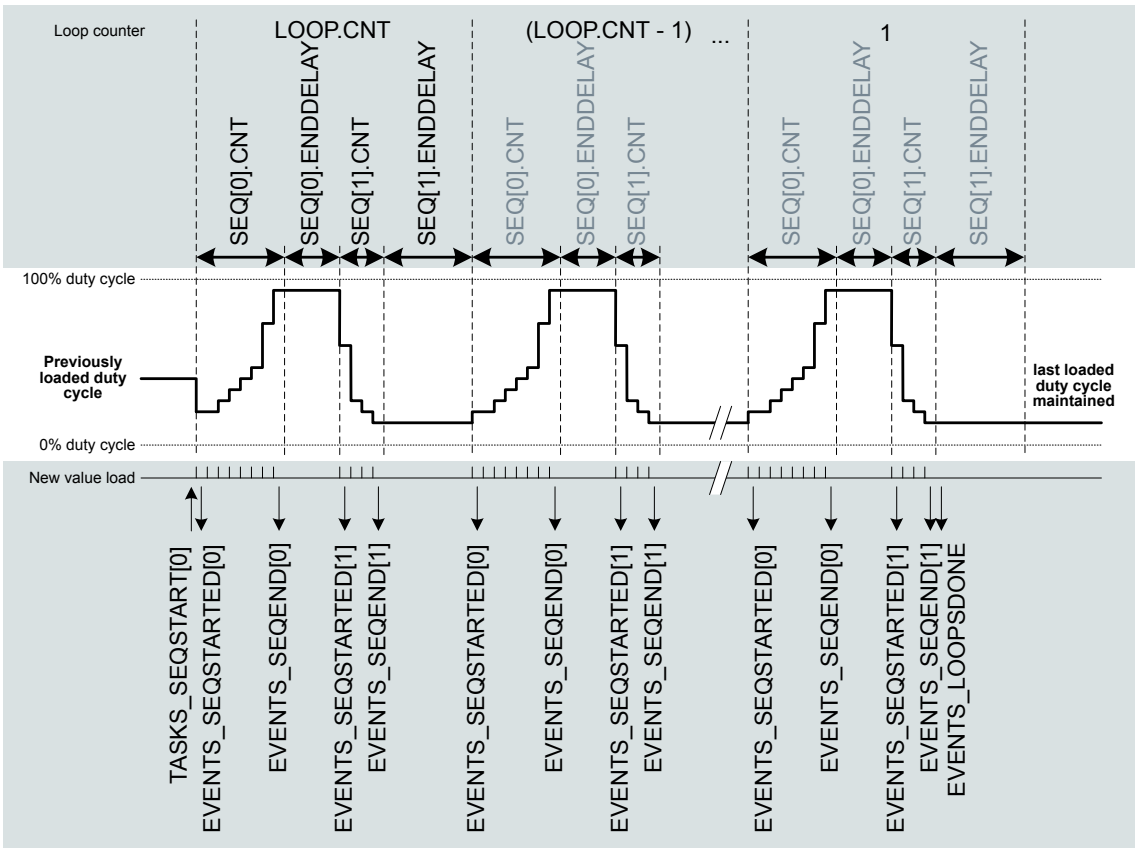

Figure 114: Complex sequence (LOOP.CNT>0) starting with SEQ[0]

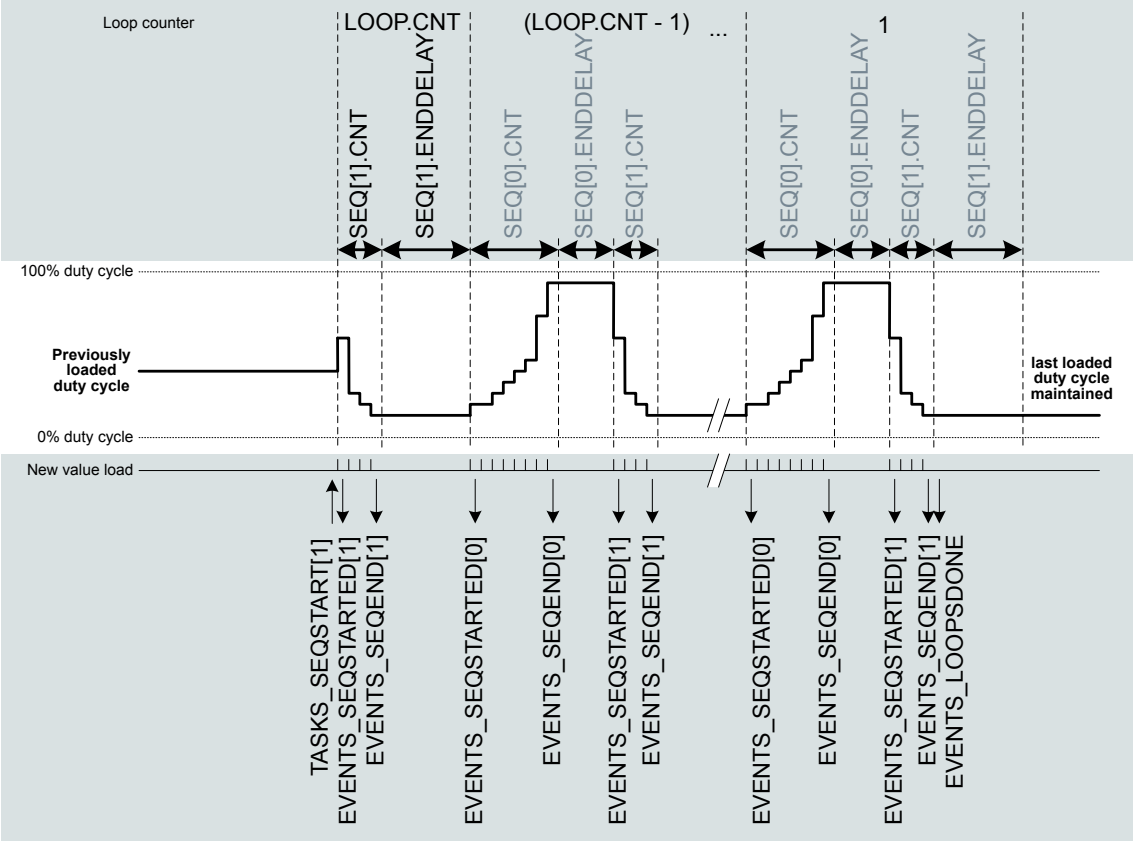

Figure 115: Complex sequence (LOOP.CNT>0) starting with SEQ[1]

Note: If a sequence is in use in a simple or complex sequence, it must have a length of SEQ[n].CNT > 0.

This example shows how the PWM module can be configured to repeat a single sequence until stopped.

```
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
// Enable the shortcut from LOOPSDONE event to SEQSTART1 task for infinite loop
NRF_PWM0->SHORTS      = (PWM_SHORTS_LOOPSDONE_SEQSTART1_Enabled <<
                                            PWM_SHORTS_LOOPSDONE_SEQSTART1_Pos);
// LOOP_CNT must be greater than 0 for the LOOPSDONE event to trigger and enable looping
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);
// To repeat a single sequence until stopped, it must be configured in SEQ[1]
NRF_PWM0->SEQ[1].PTR  = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[1].CNT  = ((sizeof(seq0_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[1] = 1;
```

7.23.3 Limitations

The previous compare value is repeated if the PWM period is shorter than the time it takes for the EasyDMA to retrieve from RAM and update the internal compare registers. This is to ensure a glitch-free operation even for very short PWM periods.

Only SEQ[1] can trigger the LOOPSDONE event upon completion, not SEQ[0]. This requires looping to be enabled (LOOP > 0) and SEQ[1].CNT > 0 when sequence playback starts.

7.23.4 Pin configuration

The OUT[n] (n=0..3) signals associated with each PWM channel are mapped to physical pins according to the configuration of PSEL.OUT[n] registers. If PSEL.OUT[n].CONNECT is set to Disconnected, the associated PWM module signal will not be connected to any physical pins.

The PSEL.OUT[n] registers and their configurations are used as long as the PWM module is enabled and the PWM generation active (wave counter started). They are retained only as long as the device is in System ON mode (see the POWER section for more information about power modes).

To ensure correct behavior in the PWM module, the pins that are used must be configured in the GPIO peripheral in the following way before the PWM module is enabled:

PWM signal	PWM pin	Direction	Output value	Comment
OUT[n]	As specified in PSEL.OUT[n] (n=0..3)	Output	0	Idle state defined in GPIO OUT register

Table 111: Recommended GPIO configuration before starting PWM generation

The idle state of a pin is defined by the OUT register in the GPIO module, to ensure that the pins used by the PWM module are driven correctly. If PWM generation is stopped by triggering a STOP task, the PWM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected pins (I/Os) for as long as the PWM module is supposed to be connected to an external PWM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

7.23.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50021000 0x40021000	APPLICATION	PWM	PWM0 : S PWM0 : NS	US	SA	Pulse width modulation unit 0	
0x50022000 0x40022000	APPLICATION	PWM	PWM1 : S PWM1 : NS	US	SA	Pulse width modulation unit 1	
0x50023000 0x40023000	APPLICATION	PWM	PWM2 : S PWM2 : NS	US	SA	Pulse width modulation unit 2	
0x50024000 0x40024000	APPLICATION	PWM	PWM3 : S PWM3 : NS	US	SA	Pulse width modulation unit 3	

Table 112: Instances

Register	Offset	Security	Description
TASKS_STOP	0x004		Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback

Register	Offset	Security	Description
TASKS_SEQSTART[n]	0x008		Loads the first PWM value on all enabled channels from sequence n, and starts playing that sequence at the rate defined in SEQ[n]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
TASKS_NEXTSTEP	0x010		Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
SUBSCRIBE_SEQSTART[n]	0x088		Subscribe configuration for task SEQSTART[n]
SUBSCRIBE_NEXTSTEP	0x090		Subscribe configuration for task NEXTSTEP
EVENTS_STOPPED	0x104		Response to STOP task, emitted when PWM pulses are no longer generated
EVENTS_SEQSTARTED[n]	0x108		First PWM period started on sequence n
EVENTS_SEQEND[n]	0x110		Emitted at end of every sequence n, when last value from RAM has been applied to wave counter
EVENTS_PWMPERIODEND	0x118		Emitted at the end of each PWM period
EVENTS_LOOPSDONE	0x11C		Concatenated sequences have been played the amount of times defined in LOOP.CNT
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_SEQSTARTED[n]	0x188		Publish configuration for event SEQSTARTED[n]
PUBLISH_SEQEND[n]	0x190		Publish configuration for event SEQEND[n]
PUBLISH_PWMPERIODEND	0x198		Publish configuration for event PWMPERIODEND
PUBLISH_LOOPSDONE	0x19C		Publish configuration for event LOOPSDONE
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		PWM module enable register
MODE	0x504		Selects operating mode of the wave counter
COUNTERTOP	0x508		Value up to which the pulse generator counter counts
PRESCALER	0x50C		Configuration for PWM_CLK
DECODER	0x510		Configuration of the decoder
LOOP	0x514		Number of playbacks of a loop
SEQ[n].PTR	0x520		Beginning address in RAM of this sequence
SEQ[n].CNT	0x524		Number of values (duty cycles) in this sequence
SEQ[n].REFRESH	0x528		Number of additional PWM periods between samples loaded into compare register
SEQ[n].ENDDELAY	0x52C		Time added after the sequence
PSEL.OUT[n]	0x560		Output pin select for PWM channel n

Table 113: Register overview

7.23.5.1 TASKS_STOP

Address offset: 0x004

Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback

Bit number																															
																															31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback
			Trigger	1	Trigger task

7.23.5.2 TASKS_SEQSTART[n] (n=0..1)

Address offset: 0x008 + (n × 0x4)

Loads the first PWM value on all enabled channels from sequence n, and starts playing that sequence at the rate defined in SEQ[n]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_SEQSTART			Loads the first PWM value on all enabled channels from sequence n, and starts playing that sequence at the rate defined in SEQ[n]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
			Trigger	1	Trigger task

7.23.5.3 TASKS_NEXTSTEP

Address offset: 0x010

Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_NEXTSTEP			Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.
			Trigger	1	Trigger task

7.23.5.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.23.5.5 SUBSCRIBE_SEQSTART[n] (n=0..1)

Address offset: 0x088 + (n × 0x4)

Subscribe configuration for task SEQSTART[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SEQSTART[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.23.5.6 SUBSCRIBE_NEXTSTEP

Address offset: 0x090

Subscribe configuration for task NEXTSTEP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task NEXTSTEP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.23.5.7 EVENTS_STOPPED

Address offset: 0x104

Response to STOP task, emitted when PWM pulses are no longer generated

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			Response to STOP task, emitted when PWM pulses are no longer generated
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.23.5.8 EVENTS_SEQSTARTED[n] (n=0..1)

Address offset: 0x108 + (n × 0x4)

First PWM period started on sequence n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_SEQSTARTED			First PWM period started on sequence n
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.23.5.9 EVENTS_SEQEND[n] (n=0..1)

Address offset: 0x110 + (n × 0x4)

Emitted at end of every sequence n, when last value from RAM has been applied to wave counter

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SEQEND			Emitted at end of every sequence n, when last value from RAM has been applied to wave counter
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.23.5.10 EVENTS_PWMPERIODEND

Address offset: 0x118

Emitted at the end of each PWM period

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_PWMPERIODEND			Emitted at the end of each PWM period
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.23.5.11 EVENTS_LOOPSDONE

Address offset: 0x11C

Concatenated sequences have been played the amount of times defined in LOOP.CNT

This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_LOOPSDONE			Concatenated sequences have been played the amount of times defined in LOOP.CNT This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.23.5.12 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.23.5.13 PUBLISH_SEQSTARTED[n] (n=0..1)

Address offset: 0x188 + (n × 0x4)

Publish configuration for event SEQSTARTED[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SEQSTARTED[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.23.5.14 PUBLISH_SEQEND[n] (n=0..1)

Address offset: 0x190 + (n × 0x4)

Publish configuration for event SEQEND[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SEQEND[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.23.5.15 PUBLISH_PWMPERIODEND

Address offset: 0x198

Publish configuration for event PWMPERIODEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event PWMPERIODEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.23.5.16 PUBLISH_LOOPSDONE

Address offset: 0x19C

Publish configuration for event LOOPSDONE

This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event LOOPSDONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.23.5.17 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SEQEND0_STOP			Shortcut between event SEQEND[0] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	SEQEND1_STOP			Shortcut between event SEQEND[1] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LOOPSDONE_SEQSTART0			Shortcut between event LOOPSDONE and task SEQSTART[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LOOPSDONE_SEQSTART1			Shortcut between event LOOPSDONE and task SEQSTART[1]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LOOPSDONE_STOP			Shortcut between event LOOPSDONE and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.23.5.18 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
B	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
C-D	RW	SEQSTARTED[i] (i=0..1)			Enable or disable interrupt for event SEQSTARTED[i]
			Disabled	0	Disable
			Enabled	1	Enable
E-F	RW	SEQEND[i] (i=0..1)			Enable or disable interrupt for event SEQEND[i]
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	PWMPERIODEND			Enable or disable interrupt for event PWMPERIODEND
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	LOOPSDONE			Enable or disable interrupt for event LOOPSDONE This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.
			Disabled	0	Disable
			Enabled	1	Enable

7.23.5.19 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
B	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C-D	RW	SEQSTARTED[i] (i=0..1)			Write '1' to enable interrupt for event SEQSTARTED[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E-F	RW	SEQEND[i] (i=0..1)			Write '1' to enable interrupt for event SEQEND[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	PWMPERIODEND			Write '1' to enable interrupt for event PWMPERIODEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	LOOPSDONE			Write '1' to enable interrupt for event LOOPSDONE This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.23.5.20 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
B	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C-D	RW	SEQSTARTED[i] (i=0..1)			Write '1' to disable interrupt for event SEQSTARTED[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E-F	RW	SEQEND[i] (i=0..1)			Write '1' to disable interrupt for event SEQEND[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	PWMPERIODEND			Write '1' to disable interrupt for event PWMPERIODEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	LOOPSDONE			Write '1' to disable interrupt for event LOOPSDONE This event triggers after the last SEQ[1] completion of the loop, and only if looping was enabled (LOOP > 0) when the sequence playback was started.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.23.5.21 ENABLE

Address offset: 0x500

PWM module enable register

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

7.23.5.22 MODE

Address offset: 0x504

Selects operating mode of the wave counter

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	UPDOWN			Selects up mode or up-and-down mode for the counter
			Up	0	Up counter, edge-aligned PWM duty cycle
			UpAndDown	1	Up and down counter, center-aligned PWM duty cycle

7.23.5.23 COUNTERTOP

Address offset: 0x508

Value up to which the pulse generator counter counts

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x000003FF					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	COUNTERTOP		[3..32767]	Value up to which the pulse generator counter counts. This register is ignored when DECODER.MODE=WaveForm and only values from RAM are used.

7.23.5.24 PRESCALER

Address offset: 0x50C

Configuration for PWM_CLK

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PRESCALER			Prescaler of PWM_CLK
			DIV_1	0	Divide by 1 (16 MHz)
			DIV_2	1	Divide by 2 (8 MHz)
			DIV_4	2	Divide by 4 (4 MHz)
			DIV_8	3	Divide by 8 (2 MHz)
			DIV_16	4	Divide by 16 (1 MHz)
			DIV_32	5	Divide by 32 (500 kHz)
			DIV_64	6	Divide by 64 (250 kHz)
			DIV_128	7	Divide by 128 (125 kHz)

7.23.5.25 DECODER

Address offset: 0x510

Configuration of the decoder

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LOAD			How a sequence is read from RAM and spread to the compare register
			Common	0	1st half word (16-bit) used in all PWM channels 0..3
			Grouped	1	1st half word (16-bit) used in channel 0..1; 2nd word in channel 2..3
			Individual	2	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in ch.3
			WaveForm	3	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in COUNTERTOP
B	RW	MODE			Selects source for advancing the active sequence
			RefreshCount	0	SEQ[n].REFRESH is used to determine loading internal compare registers
			NextStep	1	NEXTSTEP task causes a new value to be loaded to internal compare registers

7.23.5.26 LOOP

Address offset: 0x514

Number of playbacks of a loop

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CNT			Number of playbacks of pattern cycles
			Disabled	0	Looping disabled (stop at the end of the sequence)

7.23.5.27 SEQ[n].PTR (n=0..1)

Address offset: 0x520 + (n × 0x20)

Beginning address in RAM of this sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	PTR			Beginning address in RAM of this sequence

Note: See the memory chapter for details about which memories are available for EasyDMA.

7.23.5.28 SEQ[n].CNT (n=0..1)

Address offset: 0x524 + (n × 0x20)

Number of values (duty cycles) in this sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CNT			Number of values (duty cycles) in this sequence
			Disabled	0	Sequence is disabled, and shall not be started as it is empty

7.23.5.29 SEQ[n].REFRESH (n=0..1)

Address offset: 0x528 + (n × 0x20)

Number of additional PWM periods between samples loaded into compare register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	CNT			Number of additional PWM periods between samples loaded into compare register (load every REFRESH.CNT+1 PWM periods)
			Continuous	0	Update every PWM period

7.23.5.30 SEQ[n].ENDDELAY (n=0..1)

Address offset: 0x52C + (n × 0x20)

Time added after the sequence

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CNT			Time added after the sequence in PWM periods

7.23.5.31 PSEL.OUT[n] (n=0..3)

Address offset: 0x560 + (n × 0x4)

Output pin select for PWM channel n

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

7.24 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.

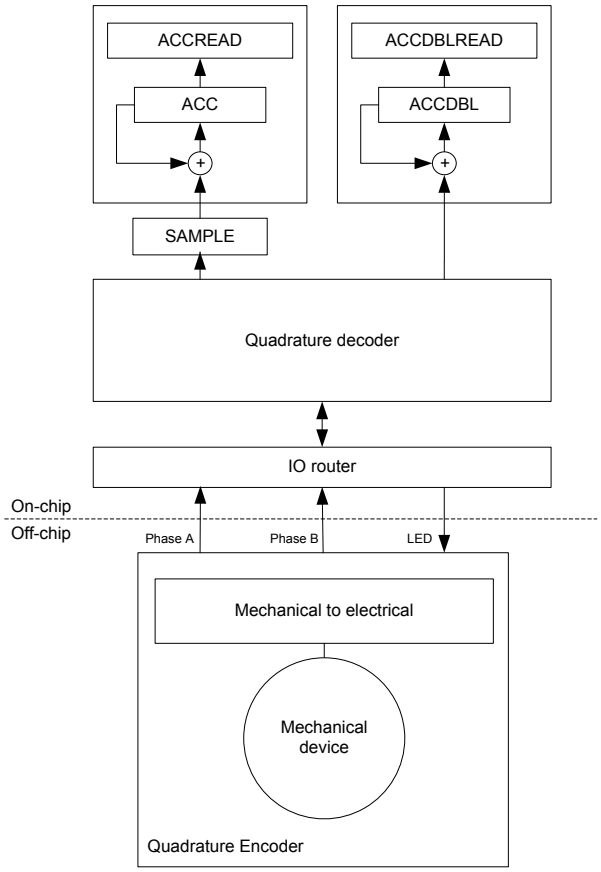

Figure 116: Quadrature decoder configuration

7.24.1 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 the waveform that changes level first. Invalid transitions may occur, meaning the two waveforms simultaneously switch. 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 behavior.

It is good practice to only change registers LEDPOL, REPORTPER, DBFEN, and LEDPRE when the QDEC is stopped.

When started, the decoder continuously samples the two input waveforms and decodes these by comparing the current sample pair (n) with the previous sample pair (n-1).

The decoding of the sample pairs is described in the table below.

Previous sample pair(n - 1)		Current samples pair(n)		SAMPLE register	ACC operation	ACCDBL operation	Description
A	B	A	B				
0	0	0	0	0	No change	No change	No movement
0	0	0	1	1	Increment	No change	Movement in positive direction
0	0	1	0	-1	Decrement	No change	Movement in negative direction
0	0	1	1	2	No change	Increment	Error: Double transition
0	1	0	0	-1	Decrement	No change	Movement in negative direction
0	1	0	1	0	No change	No change	No movement
0	1	1	0	2	No change	Increment	Error: Double transition
0	1	1	1	1	Increment	No change	Movement in positive direction
1	0	0	0	1	Increment	No change	Movement in positive direction
1	0	0	1	2	No change	Increment	Error: Double transition
1	0	1	0	0	No change	No change	No movement
1	0	1	1	-1	Decrement	No change	Movement in negative direction
1	1	0	0	2	No change	Increment	Error: Double transition
1	1	0	1	-1	Decrement	No change	Movement in negative direction
1	1	1	0	1	Increment	No change	Movement in positive direction
1	1	1	1	0	No change	No change	No movement

Table 114: Sampled value encoding

7.24.2 LED output

The LED output follows the sample period. The LED is switched on for a set period before sampling and then switched off immediately after. 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.

When using off-chip mechanical encoders not requiring an 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 pin for the LED output.

7.24.3 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). 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. Any signal with a steady state shorter than SAMPLEPER will always be suppressed by the filter. It 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.

When the debounce filters are enabled, displacements reported by the QDEC peripheral are delayed by one SAMPLEPER period.

7.24.4 Accumulators

The quadrature decoder contains two accumulator registers, ACC and ACCDBL. These registers accumulate valid motion sample values and the number of detected invalid samples (double transitions), respectively.

The ACC register accumulates 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 can fetch data when necessary instead of reading all SAMPLE register output. The ACC register holds the relative movement of the external mechanical device from 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 is 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 that do not cause the ACC to overflow or underflow will still be accepted.

The accumulator ACCDBL accumulates the number of detected double transitions since the previous clearing of the ACCDBL register.

The ACC and ACCDBL registers can be cleared by the READCLRACC and subsequently read using the ACCREAD and ACCDBLREAD registers.

The ACC register can be separately cleared by the RDCLRACC and subsequently read using the ACCREAD registers.

The ACCDBL register can be separately cleared by the RDCLRDBL and subsequently read using the ACCDBLREAD registers.

The REPORTPER register allows automated capture of multiple samples before sending an event. When a non-null displacement is captured and accumulated, a REPORTRDY event is sent. When one or more double-displacements are captured and accumulated, a DBLRDY event is sent. The REPORTPER field in this register determines how many samples must be accumulated before the contents are evaluated and a REPORTRDY or DBLRDY event is sent.

Using the RDCLRACC task (manually sent upon receiving the event, or using the DBLRDY_RDCLRACC shortcut), ACCREAD can then be read.

When a 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.

7.24.5 Output/input pins

The QDEC uses a three-pin interface to the off-chip quadrature encoder.

These pins are 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 used for the QDEC are selected using the PSEL.n registers.

7.24.6 Pin configuration

The Phase A, Phase B, and LED signals are mapped to physical pins according to the configuration specified in the PSEL.A, PSEL.B, and PSEL.LED registers respectively.

If the CONNECT field value 'Disconnected' is specified in any of these registers, the associated signal will not be connected to any physical pin. The PSEL.A, PSEL.B, and PSEL.LED registers and their configurations are only used as long as the QDEC is enabled, and retained only as long as the device is in ON mode.

When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register.

To secure correct behavior in the QDEC, the pins used by the QDEC must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 377 before enabling the QDEC. This configuration must be retained in the GPIO for the selected I/Os as long as the QDEC is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

QDEC signal	QDEC pin	Direction	Output value	Comment
Phase A	As specified in PSEL.A	Input	Not applicable	
Phase B	As specified in PSEL.B	Input	Not applicable	
LED	As specified in PSEL.LED	Input	Not applicable	

Table 115: GPIO configuration before enabling peripheral

7.24.7 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50033000 0x40033000	APPLICATION	QDEC	QDEC0 : S QDEC0 : NS	US	NA	Quadrature decoder 0	
0x50034000 0x40034000	APPLICATION	QDEC	QDEC1 : S QDEC1 : NS	US	NA	Quadrature decoder 1	

Table 116: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Task starting the quadrature decoder
TASKS_STOP	0x004		Task stopping the quadrature decoder
TASKS_READCLRACC	0x008		Read and clear ACC and ACCDBL
TASKS_RDCLRACC	0x00C		Read and clear ACC
TASKS_RDCLRDBL	0x010		Read and clear ACCDBL
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
SUBSCRIBE_READCLRACC	0x088		Subscribe configuration for task READCLRACC
SUBSCRIBE_RDCLRACC	0x08C		Subscribe configuration for task RDCLRACC
SUBSCRIBE_RDCLRDBL	0x090		Subscribe configuration for task RDCLRDBL
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
PUBLISH_SAMPLERDY	0x180		Publish configuration for event SAMPLERDY
PUBLISH_REPORTRDY	0x184		Publish configuration for event REPORTRDY
PUBLISH_ACCOF	0x188		Publish configuration for event ACCOF
PUBLISH_DBLRDY	0x18C		Publish configuration for event DBLRDY
PUBLISH_STOPPED	0x190		Publish configuration for event STOPPED
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		Enable the quadrature decoder
LEDPOL	0x504		LED output pin polarity

Register	Offset	Security	Description
SAMPLEPER	0x508		Sample period
SAMPLE	0x50C		Motion sample value
REPORTPER	0x510		Number of samples to be taken before REPORTRDY and DBLRDY events can be generated
ACC	0x514		Register accumulating the valid transitions
ACCREAD	0x518		Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task
PSEL.LED	0x51C		Pin select for LED signal
PSEL.A	0x520		Pin select for A signal
PSEL.B	0x524		Pin select for B signal
DBFEN	0x528		Enable input debounce filters
LEDPRE	0x540		Time period the LED is switched ON prior to sampling
ACCDBL	0x544		Register accumulating the number of detected double transitions
ACCDBLREAD	0x548		Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Table 117: Register overview

7.24.7.1 TASKS_START

Address offset: 0x000

Task starting the quadrature decoder

When started, the SAMPLE register will be continuously updated at the rate given in the SAMPLEPER register.

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Task starting the quadrature decoder When started, the SAMPLE register will be continuously updated at the rate given in the SAMPLEPER register.
			Trigger	1	Trigger task

7.24.7.2 TASKS_STOP

Address offset: 0x004

Task stopping the quadrature decoder

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Task stopping the quadrature decoder
			Trigger	1	Trigger task

7.24.7.3 TASKS_READCLRACC

Address offset: 0x008

Read and clear ACC and ACCDBL

Task transferring the content of ACC to ACCREAD and the content of ACCDBL to ACCDBLREAD, and then clearing the ACC and ACCDBL registers. These read-and-clear operations will be done atomically.

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_READCLRACC			Read and clear ACC and ACCDBL
					Task transferring the content of ACC to ACCREAD and the content of ACCDBL to ACCDBLREAD, and then clearing the ACC and ACCDBL registers. These read-and-clear operations will be done atomically.
			Trigger	1	Trigger task

7.24.7.4 TASKS_RDCLRACC

Address offset: 0x00C

Read and clear ACC

Task transferring the content of ACC to ACCREAD, and then clearing the ACC register. This read-and-clear operation will be done atomically.

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RDCLRACC			Read and clear ACC
					Task transferring the content of ACC to ACCREAD, and then clearing the ACC register. This read-and-clear operation will be done atomically.
			Trigger	1	Trigger task

7.24.7.5 TASKS_RDCLRDBL

Address offset: 0x010

Read and clear ACCDBL

Task transferring the content of ACCDBL to ACCDBLREAD, and then clearing the ACCDBL register. This read-and-clear operation will be done atomically.

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RDCLRDBL			Read and clear ACCDBL
					Task transferring the content of ACCDBL to ACCDBLREAD, and then clearing the ACCDBL register. This read-and-clear operation will be done atomically.
			Trigger	1	Trigger task

7.24.7.6 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task START

When started, the SAMPLE register will be continuously updated at the rate given in the SAMPLEPER 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.24.7.7 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.24.7.8 SUBSCRIBE_READCLRACC

Address offset: 0x088

Subscribe configuration for task READCLRACC

Task transferring the content of ACC to ACCREAD and the content of ACCDBL to ACCDBLREAD, and then clearing the ACC and ACCDBL registers. These read-and-clear operations will be done atomically.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task READCLRACC will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.24.7.9 SUBSCRIBE_RDCLRACC

Address offset: 0x08C

Subscribe configuration for task RDCLRACC

Task transferring the content of ACC to ACCREAD, and then clearing the ACC register. This read-and-clear operation will be done atomically.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RDCLRACC will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.24.7.10 SUBSCRIBE_RDCLRDBL

Address offset: 0x090

Subscribe configuration for task RDCLRDBL

Task transferring the content of ACCDBL to ACCDBLREAD, and then clearing the ACCDBL register. This read-and-clear operation will be done atomically.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RDCLRDBL will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.24.7.11 EVENTS_SAMPLERDY

Address offset: 0x100

Event being generated for every new sample value written to the SAMPLE register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SAMPLERDY			Event being generated for every new sample value written to the SAMPLE register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.24.7.12 EVENTS_REPORTRDY

Address offset: 0x104

Non-null report ready

Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_REPORTRDY			Non-null report ready
					Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.24.7.13 EVENTS_ACCOF

Address offset: 0x108

ACC or ACCDBL register overflow

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ACCOF			ACC or ACCDBL register overflow
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.24.7.14 EVENTS_DBLRDY

Address offset: 0x10C

Double displacement(s) detected

Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DBLRDY			Double displacement(s) detected
					Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.24.7.15 EVENTS_STOPPED

Address offset: 0x110

QDEC has been stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			QDEC has been stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.24.7.16 PUBLISH_SAMPLERDY

Address offset: 0x180

Publish configuration for event SAMPLERDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SAMPLERDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.24.7.17 PUBLISH_REPORTRDY

Address offset: 0x184

Publish configuration for event REPORTRDY

Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event REPORTRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.24.7.18 PUBLISH_ACCOF

Address offset: 0x188

Publish configuration for event ACCOF

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ACCOF will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.24.7.19 PUBLISH_DBLRDY

Address offset: 0x18C

Publish configuration for event DBLRDY

Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DBLRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.24.7.20 PUBLISH_STOPPED

Address offset: 0x190

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.24.7.21 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	REPORTRDY_READCLRACC			Shortcut between event REPORTRDY and task READCLRACC

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	SAMPLERDY_STOP			Shortcut between event SAMPLERDY and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	REPORTRDY_RDCLRACC			Shortcut between event REPORTRDY and task RDCLRACC
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	REPORTRDY_STOP			Shortcut between event REPORTRDY and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	DBLRDY_RDCLRDBL			Shortcut between event DBLRDY and task RDCLRDBL
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	DBLRDY_STOP			Shortcut between event DBLRDY and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G	RW	SAMPLERDY_READCLRACC			Shortcut between event SAMPLERDY and task READCLRACC
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.24.7.22 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SAMPLERDY			Write '1' to enable interrupt for event SAMPLERDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REPORTRDY			Write '1' to enable interrupt for event REPORTRDY Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACCOF			Write '1' to enable interrupt for event ACCOF
			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				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
D	RW	DBLRDY			Write '1' to enable interrupt for event DBLRDY Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.24.7.23 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SAMPLERDY			Write '1' to disable interrupt for event SAMPLERDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	REPORTRDY			Write '1' to disable interrupt for event REPORTRDY Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACCOF			Write '1' to disable interrupt for event ACCOF
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	DBLRDY			Write '1' to disable interrupt for event DBLRDY Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).
			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				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
E	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.24.7.24 ENABLE

Address offset: 0x500

Enable the quadrature decoder

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

ID	R/W	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

7.24.7.25 LEDPOL

Address offset: 0x504

LED output pin polarity

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

ID	R/W	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

7.24.7.26 SAMPLEPER

Address offset: 0x508

Sample period

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SAMPLEPER			Sample period. The SAMPLE register will be updated for every new sample
			128us	0	128 µs

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
			256us	1	256 µs
			512us	2	512 µs
			1024us	3	1024 µs
			2048us	4	2048 µs
			4096us	5	4096 µs
			8192us	6	8192 µs
			16384us	7	16384 µs
			32ms	8	32768 µs
			65ms	9	65536 µs
			131ms	10	131072 µs

7.24.7.27 SAMPLE

Address offset: 0x50C

Motion sample value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	SAMPLE		[-1..2]	Last motion sample The value is a 2's complement value, and the sign gives the direction of the motion. The value '2' indicates a double transition.

7.24.7.28 REPORTPER

Address offset: 0x510

Number of samples to be taken before REPORTRDY and DBLRDY events can be generated

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	REPORTPER			Specifies the number of samples to be accumulated in the ACC register before the REPORTRDY and DBLRDY events can be generated. The report period in [µs] is given as: RPUS = SP * RP Where RPUS is the report period in [µs/report], SP is the sample period in [µs/sample] specified in SAMPLEPER, and RP is the report period in [samples/report] specified in REPORTPER .
			10Smpl	0	10 samples/report
			40Smpl	1	40 samples/report
			80Smpl	2	80 samples/report
			120Smpl	3	120 samples/report
			160Smpl	4	160 samples/report
			200Smpl	5	200 samples/report
			240Smpl	6	240 samples/report
			280Smpl	7	280 samples/report
			1Smpl	8	1 sample/report

7.24.7.29 ACC

Address offset: 0x514

Register accumulating the valid transitions

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

7.24.7.30 ACCREAD

Address offset: 0x518

Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task

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

7.24.7.31 PSEL.LED

Address offset: 0x51C

Pin select for LED signal

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

7.24.7.32 PSEL.A

Address offset: 0x520

Pin select for A signal

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

7.24.7.33 PSEL.B

Address offset: 0x524

Pin select for B signal

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

7.24.7.34 DBFEN

Address offset: 0x528

Enable input debounce filters

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DBFEN			Enable input debounce filters
			Disabled	0	Debounce input filters disabled
			Enabled	1	Debounce input filters enabled

7.24.7.35 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LEDPRE		[1..511]	Period in µs the LED is switched on prior to sampling

7.24.7.36 ACCDBL

Address offset: 0x544

Register accumulating the number of detected double transitions

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

7.24.7.37 ACCDBLREAD

Address offset: 0x548

Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

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

ID	R/W	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.

7.24.8 Electrical specification

7.24.8.1 QDEC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{SAMPLE}	Time between sampling signals from quadrature decoder	128		131072	µs
t_{LED}	Time from LED is turned on to signals are sampled	0		511	µs

7.25 QSPI — Quad serial peripheral interface

The QSPI peripheral provides support for communicating with an external flash memory device using SPI.

The main features for the QSPI peripheral are:

- Single/dual/quad SPI input/output
- 6 to 96 MHz configurable clock frequency
- Single-word read/write access from/to external flash
- EasyDMA for block read and write transfers
- Up to 48 MB/sec EasyDMA read rate
- Execute in place (XIP) for executing program directly from external flash
- XIP access can optionally be disabled
- On-the-fly encryption and decryption, including EasyDMA and XIP

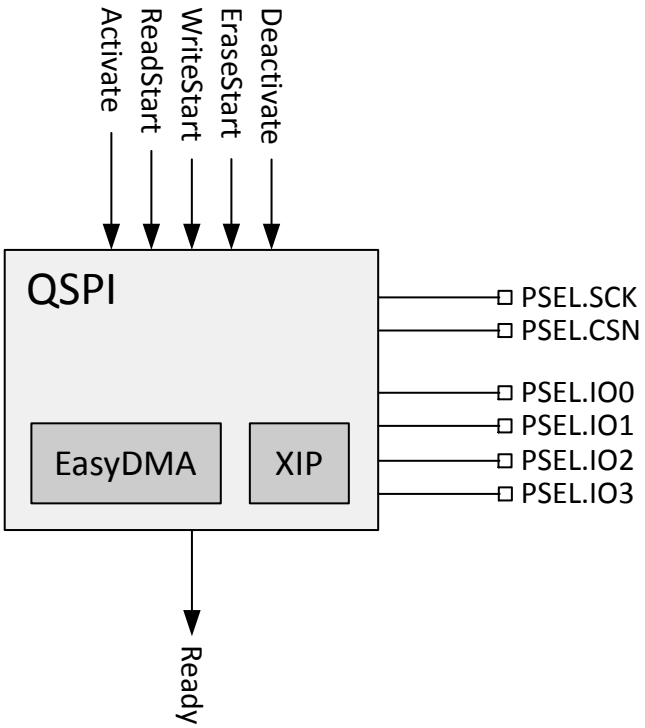

Figure 117: Block diagram

7.25.1 Configuring QSPI

Before any data can be transferred to or from the external flash memory, the peripheral needs to be configured.

1. Select the mandatory input/output pins in the following registers:
   - PSEL.SCK on page 411
   - PSEL.CSN on page 411
   - PSEL.IO0 on page 411
   - PSEL.IO1 on page 412
   - PSEL.IO2 on page 412
   - PSEL.IO3 on page 412

   For which pins to use, see Pin assignments on page 783. Only the dedicated QSPI pins shall be used.
2. To ensure stable operation, set the GPIO drive strength to high drive. See the GPIO — General purpose input/output on page 219 chapter for details on how to configure GPIO drive strength.
3. Activate the dedicated peripheral setting of the GPIO pin. See the GPIO — General purpose input/output on page 219 chapter for details on how to assign pins between cores, peripherals, or subsystems.
4. Configure the interface towards the external flash memory using IFCONFIG0 on page 413, IFCONFIG1 on page 418, and ADDRCONF on page 419.
5. Enable the QSPI peripheral and acquire I/O pins using ENABLE on page 409.
6. Activate the external flash memory interface using the ACTIVATE task. The READY event will be generated when the interface has been activated and the external flash memory is ready for access.

Note:

If the IFCONFIG0 register is configured to use the quad mode, the external flash device also needs to be set in the quad mode before any data transfers can take place.

This can be done by sending custom instructions to the external flash device, as described in Sending custom instructions on page 396.

7.25.2 Write operation

A write operation to the external flash is configured using the WRITE.DST, WRITE.SRC, and WRITE.CNT registers. It is started using the WRITESTART task.

The READY event is generated when the transfer is complete.

The QSPI peripheral automatically takes care of splitting DMA transfers into page writes.

7.25.3 Read operation

A read operation from the external flash is configured using the READ.SRC, READ.DST, and READ.CNT registers. It is started using the READSTART task.

The READY event is generated when the transfer is complete.

7.25.4 Erase operation

Erase of pages/blocks of the external flash is configured using the ERASE.PTR and ERASE.LEN registers. It is started using the ERASESTART task.

The READY event is generated when the erase operation has been started.

In this case, the READY event will not indicate that the erase operation of the flash has been completed, but it only signals that the erase operation has been started. The actual status of the erase operation can normally be read from the external flash using a custom instruction (see Sending custom instructions on page 396).

7.25.5 Execute in place

Execute in place (XIP) allows the CPU to execute program code directly from the external flash.

After the external flash has been configured, the CPU can execute code from the external flash by accessing the XIP memory region. See the following figure and Memory on page 18 for details.

The XIP memory region is read-only, writing to it will result in a bus error.

When accessing the XIP memory region, the start address of this XIP memory region will map to the address XIPOFFSET on page 413 of the external flash.

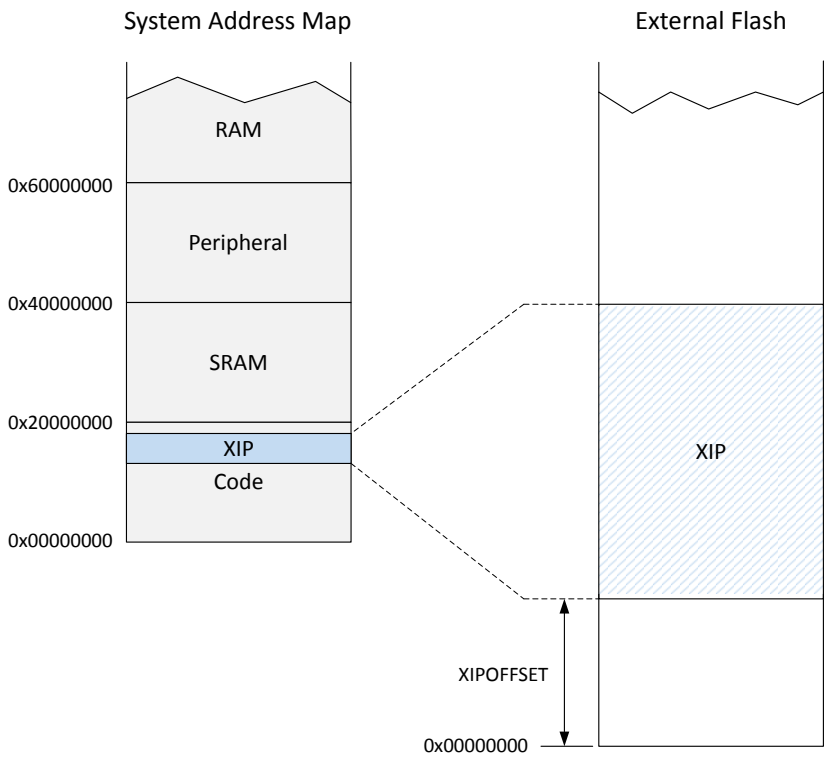

Figure 118: XIP memory map

7.25.6 Encryption

The contents of an external flash memory can be protected using stream cipher encryption. Encryption can be configured and enabled independently for XIP and EasyDMA, with separate keys and nonce.

Once configured and enabled, the stream cipher operates between the AHB bus and the external flash, encrypting and decrypting data passing through.

The following figure shows the stream cipher block with the three configuration registers. The stream cipher uses an AES 128 encryption operation to form the keystream from key, nonce, and external memory address. The keystream then combines each 32-bit plaintext digit one at a time with the corresponding digit of the keystream.

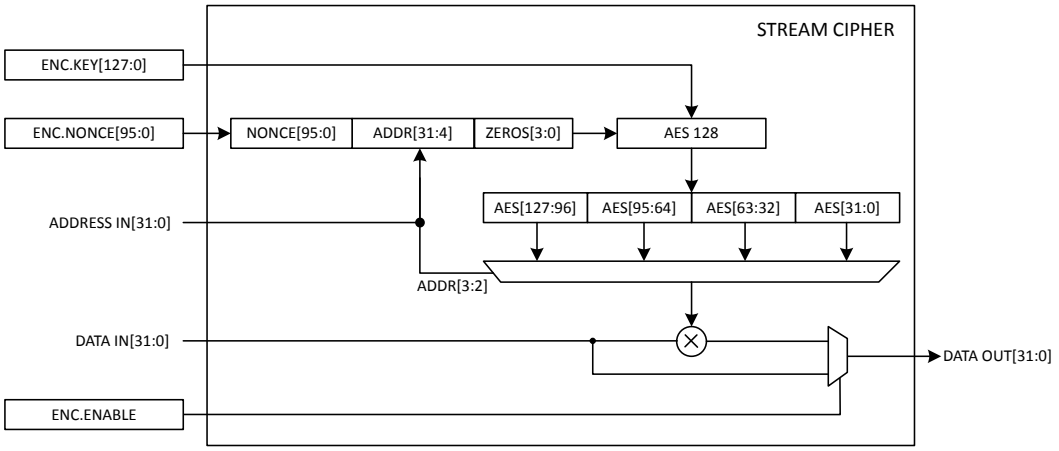

Figure 119: Stream cipher

The same nonce and key must be used for both encryption and decryption of the same memory address.

The memory address used for encryption is the external flash memory address and thus independent of XIPOFFSET on page 413. This means a second firmware image can be encrypted and written using

EasyDMA, then XIPOFFSET on page 413 set to point to the new firmware image before executing from it.

Stream ciphers are symmetric. They do not differentiate between encrypting or decrypting, reading or writing. Thus, if the contents of a plain text external flash is read when stream cipher is enabled, the data provided to the MCU is encrypted.

Execute in place (XIP)

Enable the stream cipher for QSPI XIP by doing the following steps.

1. Configure keys using XIP_ENC.KEY0 on page 414 through XIP_ENC.KEY3 on page 415.
2. Configure nonce using XIP_ENC.NONCE0 on page 415 through XIP_ENC.NONCE2 on page 415.
3. Set XIP_ENC.ENABLE on page 416.

Any instructions or data read from the XIP interface will now pass through the stream cipher.

EasyDMA

Enable the stream cipher for QSPI EasyDMA by doing the following steps.

1. Configure keys using DMA_ENC.KEY0 on page 416 through DMA_ENC.KEY3 on page 416.
2. Configure nonce using DMA_ENC.NONCE0 on page 417 through DMA_ENC.NONCE2 on page 417.
3. Set DMA_ENC.ENABLE on page 417.

Any data read from or written to the external flash over the EasyDMA interface will now pass through the stream cipher.

7.25.7 Sending custom instructions

Custom instructions can be sent to the external flash using the CINSTRCONF, CINSTRDAT0, and CINSTRDAT1 registers. It is possible to send an instruction consisting of a one-byte opcode and up to 8 bytes of additional data and to read its response.

A custom instruction is prepared by first writing the data to be sent to CINSTRDAT0 and CINSTRDAT1 before writing the opcode and other configurations to the CINSTRCONF register.

The custom instruction is sent when the CINSTRCONF register is written and it is always sent on a single data line SPI interface.

The READY event will be generated when the custom instruction has been sent.

After a custom instruction has been sent, the CINSTRDAT0 and CINSTRDAT1 will contain the response bytes from the custom instruction.

The data of custom instructions is not part of the stream cipher encryption.

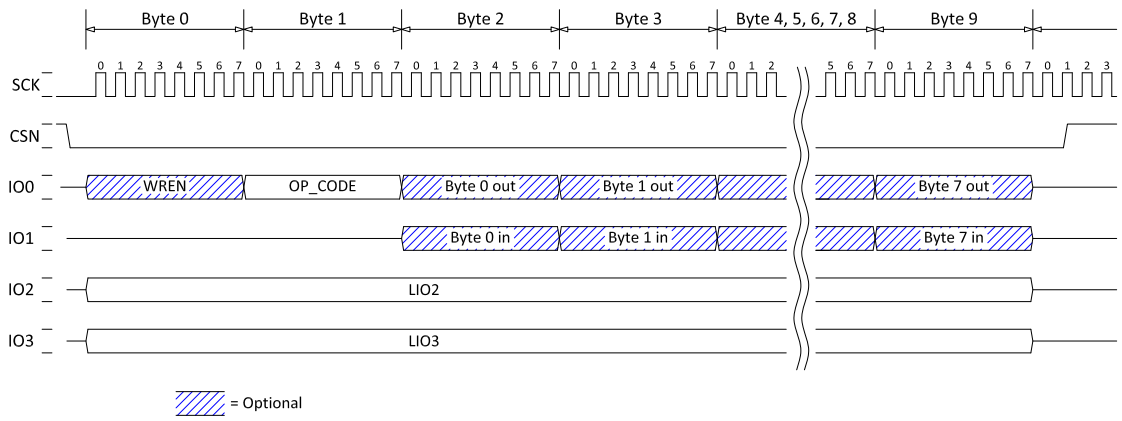

Figure 120: Sending custom instruction

7.25.7.1 Long frame mode

The LFEN and LFSTOP fields in the CINSTRCONF register control the operation of the custom instruction Long frame mode. Long frame mode is a mechanism that permits arbitrary byte length custom instructions. While in Long frame mode a long custom instruction sequence is split in multiple writes to the CINSTRDAT0 and CINSTRDAT1 registers.

To enable Long frame mode every write to the CINSTRCONF register must have the LFEN field set to `1`. The contents of the OPCODE field will be transmitted after the first write to CINSTRCONF and will be omitted in every subsequent write to this register. For subsequent writes the number of data bytes as specified in the LENGTH field are transferred (that is the value of LENGTH - 1 data bytes). The values of the LIO2 and LIO3 fields are set in the first write to CINSTRCONF and will apply for the entire custom instruction transmission until the long frame is finalized.

To finalize a long frame transmission, the LFSTOP field in the CINSTRCONF register must be set to `1` in the last write to this register.

7.25.8 Deep power-down mode

The external flash memory can be put in Deep power-down mode (DPM) to minimize its current consumption when there is no need to access the memory.

DPM is enabled in register IFCONFIG0 on page 413 and configured in register DPMDUR on page 419. The DPM status of the external memory can be read in the STATUS register. The DPMDUR register has to be configured according to the external flash specification to get the information in the STATUS register and the timing of the READY event correct.

Entering or exiting DPM is controlled using register IFCONFIG1 on page 418.

7.25.9 Instruction set

The following table shows the instruction set supported by QSPI when communicating with an external flash device.

Instruction	Opcode	Description
WREN	0x06	Write enable
RDSR	0x05	Read status register
WRSR	0x01	Write status register
FASTREAD	0x0B	Read bytes at higher speed
READ2O	0x3B	Dual-read output
READ2IO	0xBB	Dual-read input/output
READ4O	0x6B	Quad-read output
READ4IO	0xEB	Quad-read input/output
PP	0x02	Page program
PP2O	0xA2	Dual-page program output
PP4O	0x32	Quad-page program output
PP4IO	0x38	Quad-page program input/output
SE	0x20	Sector erase
BE	0xD8	Block erase
CE	0xC7	Chip erase
DP	0xB9	Enter Deep power-down mode
DPE	0xAB	Exit Deep power-down mode
EN4B	Specified in the ADDRCONF register	Enable 32 bit address mode

Table 118: Instruction set

7.25.10 Interface description

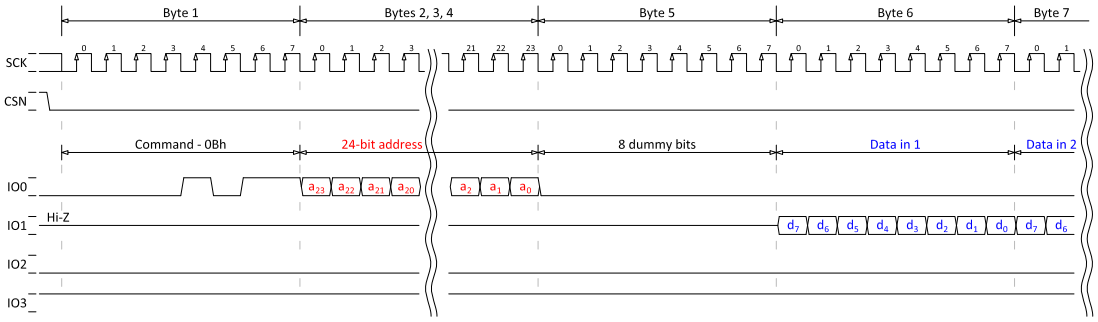

Figure 121: 24-bit FASTREAD, SPIMODE = MODE0

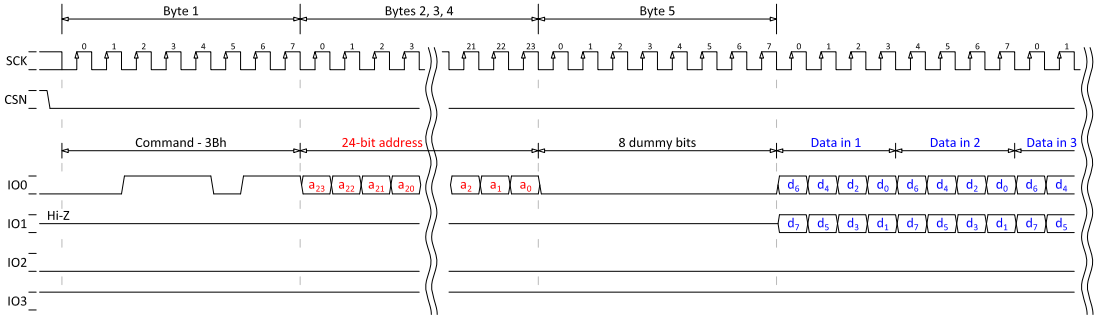

Figure 122: 24-bit READ2O (dual-read output), SPIMODE = MODE0

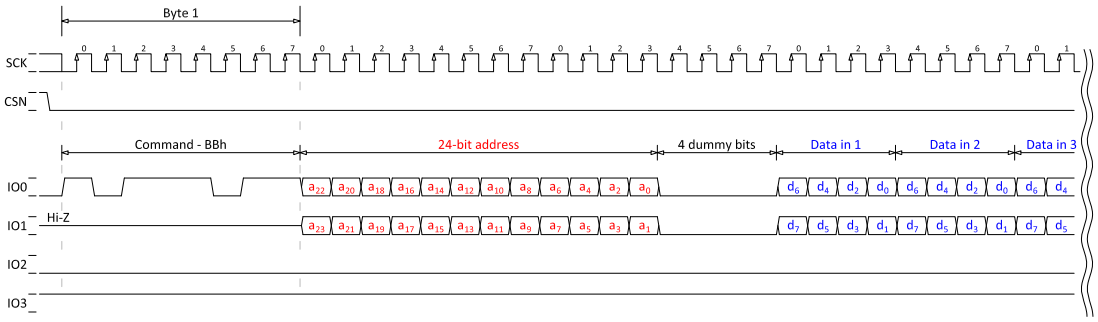

Figure 123: 24-bit READ2IO (dual read input/output), SPIMODE = MODE0

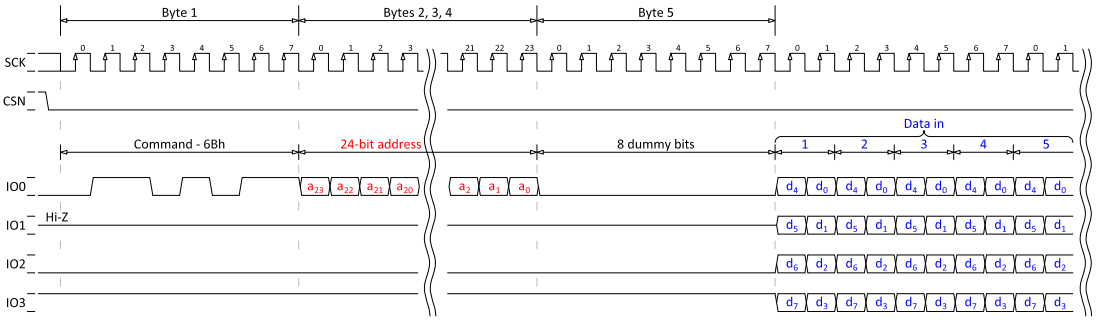

Figure 124: 24-bit READ4O (quad-read output), SPIMODE = MODE0

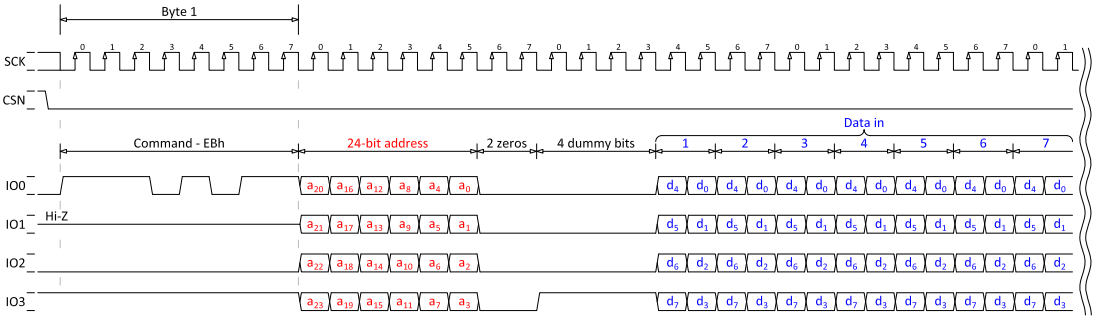

Figure 125: 24-bit READ4IO (quad-read input/output), SPIMODE = MODE0

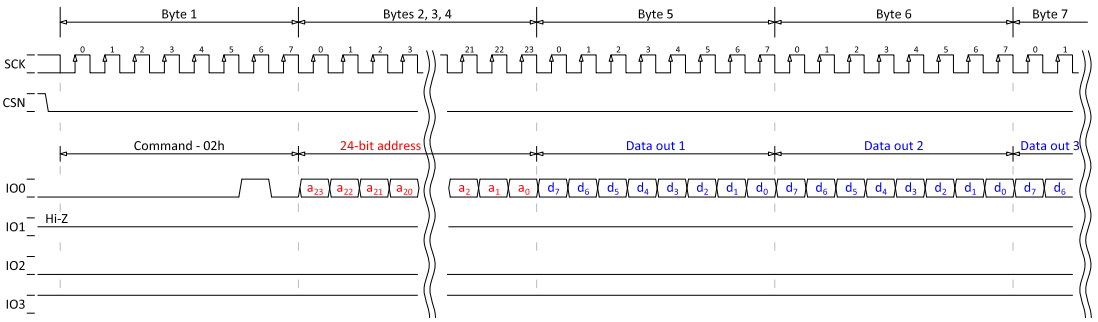

Figure 126: 24-bit PP (page program), SPIMODE = MODE0

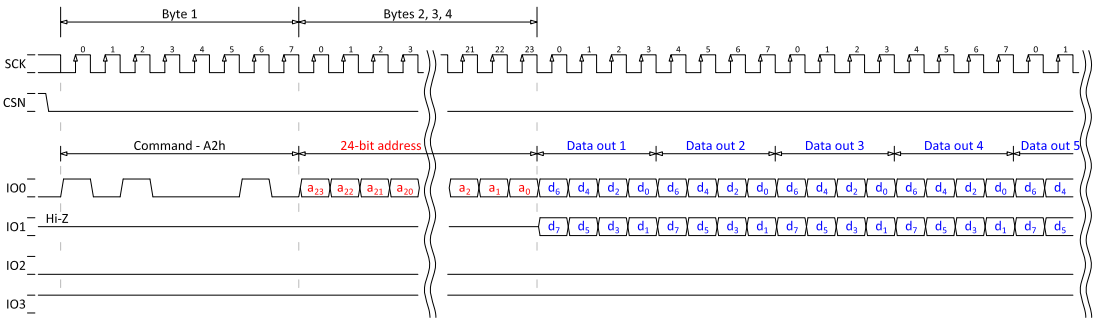

Figure 127: 24-bit PP2O (dual-page program output), SPIMODE = MODE0

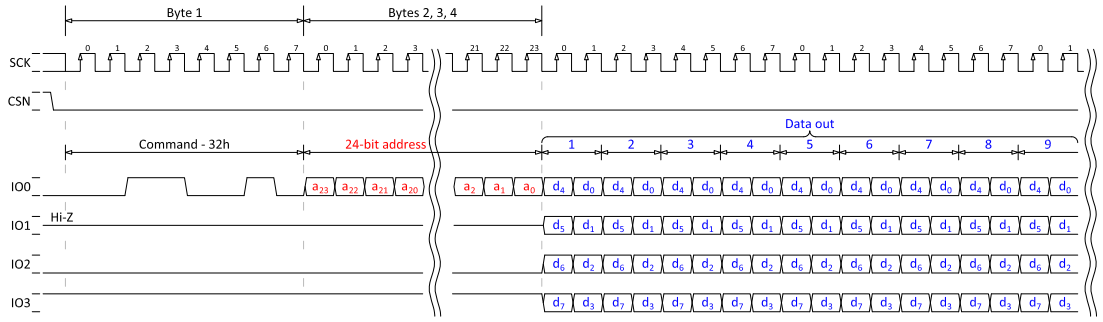

Figure 128: 24-bit PP4O (quad page program output), SPIMODE = MODE0

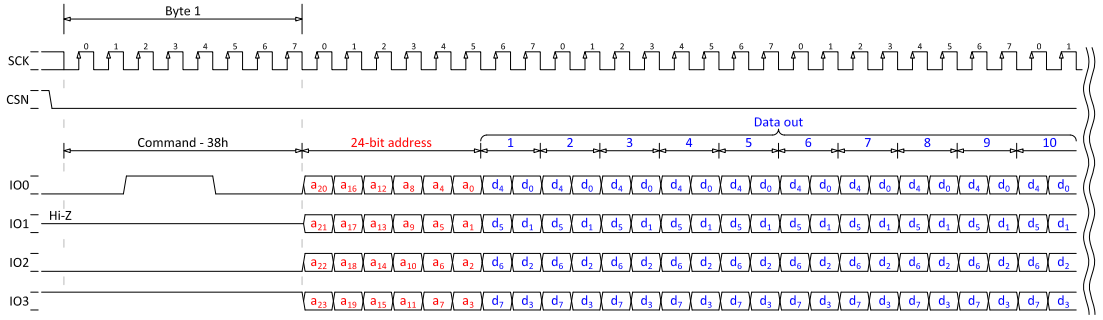

Figure 129: 24-bit PP4IO (quad page program input/output), SPIMODE = MODE0

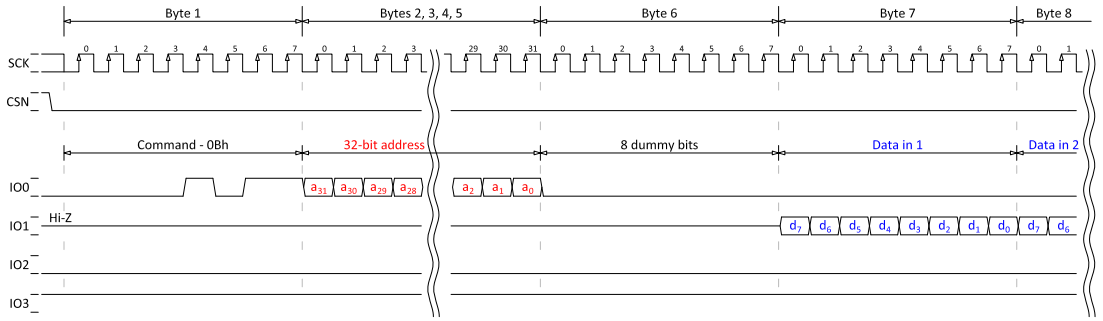

Figure 130: 32-bit FASTREAD, SPIMODE = MODE0

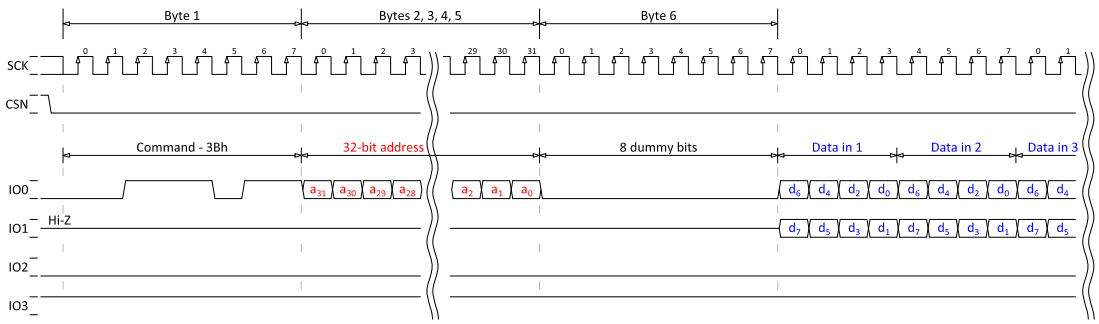

Figure 131: 32-bit READ2O (dual-read output), SPIMODE = MODE0

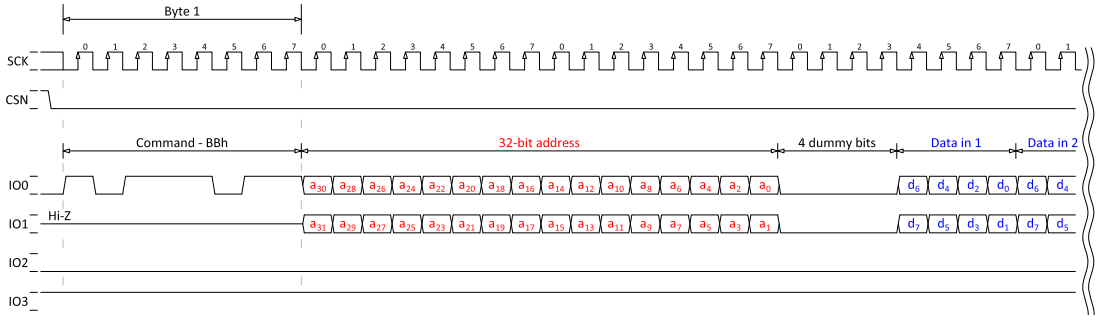

Figure 132: 32-bit READ2IO (dual read input/output), SPIMODE = MODE0

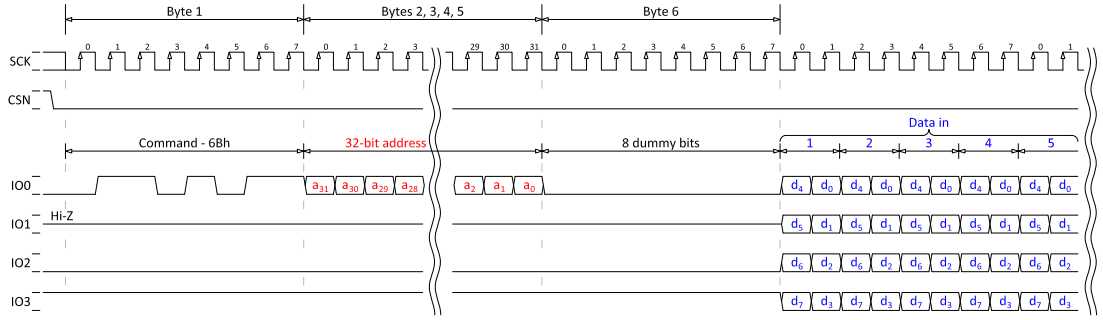

Figure 133: 32-bit READ4O (quad-read output), SPIMODE = MODE0

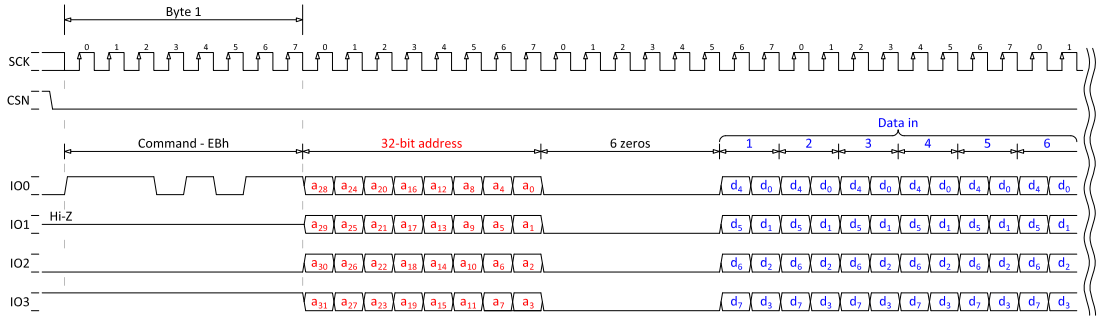

Figure 134: 32-bit READ4IO (quad-read input/output), SPIMODE = MODE0

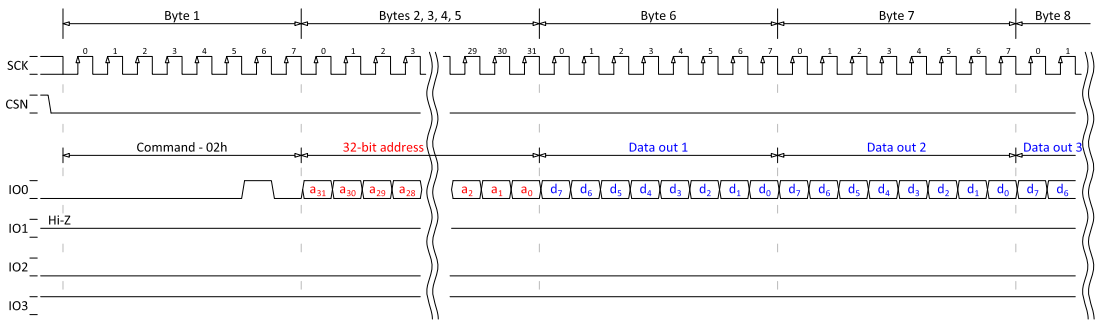

Figure 135: 32-bit PP (page program), SPIMODE = MODE0

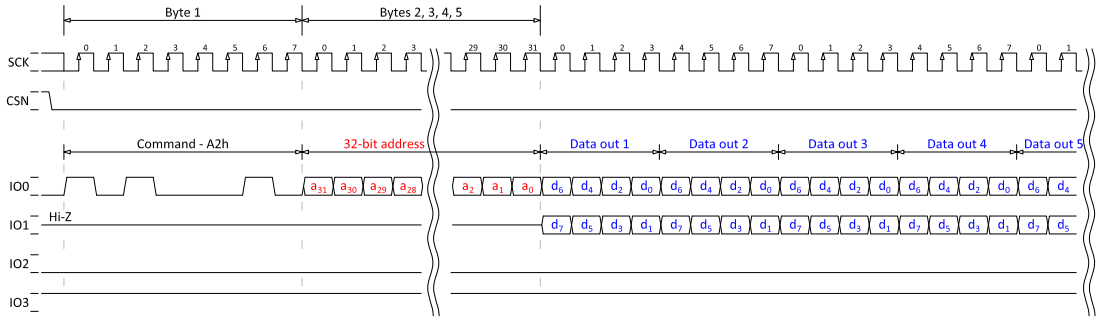

Figure 136: 32-bit PP2O (dual-page program output), SPIMODE = MODE0

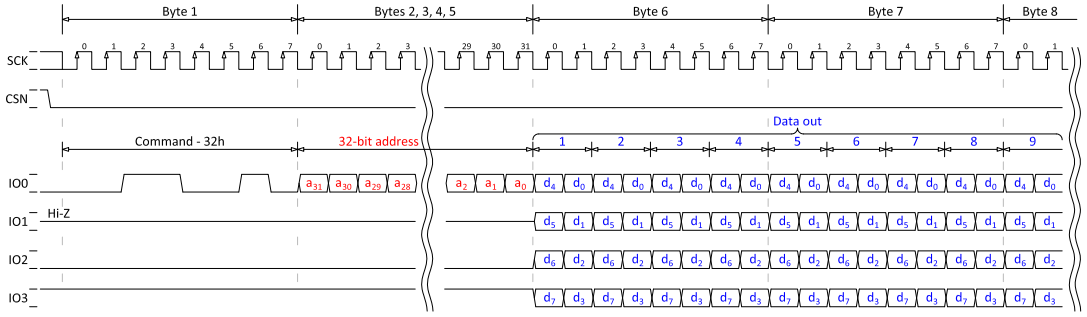

Figure 137: 32-bit PP4O (quad-page program output), SPIMODE = MODE0

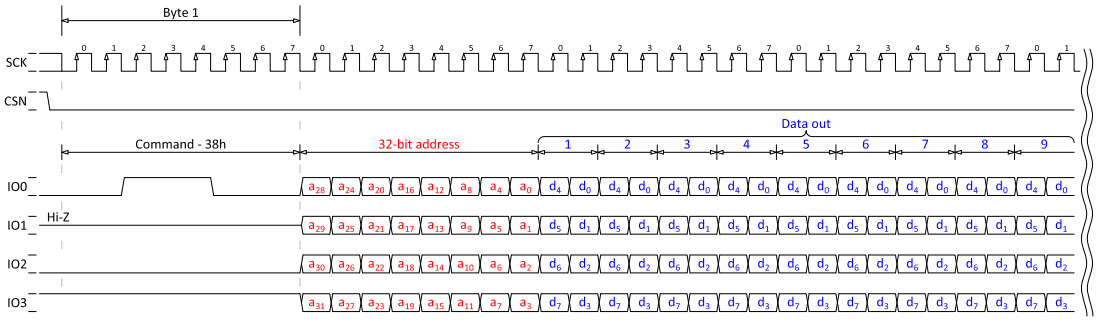

Figure 138: 32-bit PP4IO (quad page program input/output), SPIMODE = MODE0

7.25.11 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5002B000 0x4002B000	APPLICATION	QSPI	QSPI : S QSPI : NS	US	SA	External memory (quad serial peripheral) interface	Supports 192 MHz and 96 MHz PCLK192M frequency

Table 119: Instances

Register	Offset	Security	Description
TASKS_ACTIVATE	0x000		Activate QSPI interface
TASKS_READSTART	0x004		Start transfer from external flash memory to internal RAM
TASKS_WRITESTART	0x008		Start transfer from internal RAM to external flash memory
TASKS_ERASESTART	0x00C		Start external flash memory erase operation
TASKS_DEACTIVATE	0x010		Deactivate QSPI interface
SUBSCRIBE_ACTIVATE	0x080		Subscribe configuration for task ACTIVATE
SUBSCRIBE_READSTART	0x084		Subscribe configuration for task READSTART
SUBSCRIBE_WRITESTART	0x088		Subscribe configuration for task WRITESTART
SUBSCRIBE_ERASESTART	0x08C		Subscribe configuration for task ERASESTART
SUBSCRIBE_DEACTIVATE	0x090		Subscribe configuration for task DEACTIVATE
EVENTS_READY	0x100		QSPI peripheral is ready. This event will be generated as a response to all QSPI tasks except DEACTIVATE.
PUBLISH_READY	0x180		Publish configuration for event READY
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		Enable QSPI peripheral and acquire the pins selected in PSELn registers
READ.SRC	0x504		Flash memory source address
READ.DST	0x508		RAM destination address
READ.CNT	0x50C		Read transfer length
WRITE.DST	0x510		Flash destination address
WRITE.SRC	0x514		RAM source address
WRITE.CNT	0x518		Write transfer length
ERASE.PTR	0x51C		Start address of flash block to be erased
ERASE.LEN	0x520		Size of block to be erased.
PSEL.SCK	0x524		Pin select for serial clock SCK
PSEL.CSN	0x528		Pin select for chip select signal CSN.
PSEL.IO0	0x530		Pin select for serial data MOSI/IO0.
PSEL.IO1	0x534		Pin select for serial data MISO/IO1.
PSEL.IO2	0x538		Pin select for serial data WP/IO2.
PSEL.IO3	0x53C		Pin select for serial data HOLD/IO3.
XIPOFFSET	0x540		Address offset into the external memory for Execute in Place operation.
IFCONFIG0	0x544		Interface configuration.
XIPEN	0x54C		Enable Execute in Place operation.
XIP_ENC.KEY0	0x560		Bits 31:0 of XIP AES KEY
XIP_ENC.KEY1	0x564		Bits 63:32 of XIP AES KEY
XIP_ENC.KEY2	0x568		Bits 95:64 of XIP AES KEY
XIP_ENC.KEY3	0x56C		Bits 127:96 of XIP AES KEY
XIP_ENC.NONCE0	0x570		Bits 31:0 of XIP NONCE
XIP_ENC.NONCE1	0x574		Bits 63:32 of XIP NONCE
XIP_ENC.NONCE2	0x578		Bits 95:64 of XIP NONCE
XIP_ENC.ENABLE	0x57C		Enable stream cipher for XIP
DMA_ENC.KEY0	0x580		Bits 31:0 of DMA AES KEY
DMA_ENC.KEY1	0x584		Bits 63:32 of DMA AES KEY

Register	Offset	Security	Description
DMA_ENC.KEY2	0x588		Bits 95:64 of DMA AES KEY
DMA_ENC.KEY3	0x58C		Bits 127:96 of DMA AES KEY
DMA_ENC.NONCE0	0x590		Bits 31:0 of DMA NONCE
DMA_ENC.NONCE1	0x594		Bits 63:32 of DMA NONCE
DMA_ENC.NONCE2	0x598		Bits 95:64 of DMA NONCE
DMA_ENC.ENABLE	0x59C		Enable stream cipher for EasyDMA
IFCONFIG1	0x600		Interface configuration.
STATUS	0x604		Status register.
DPMDUR	0x614		Set the duration required to enter/exit deep power-down mode (DPM).
ADDRCONF	0x624		Extended address configuration.
CINSTRCONF	0x634		Custom instruction configuration register.
CINSTRDAT0	0x638		Custom instruction data register 0.
CINSTRDAT1	0x63C		Custom instruction data register 1.
IFTIMING	0x640		SPI interface timing.

Table 120: Register overview

7.25.11.1 TASKS_ACTIVATE

Address offset: 0x000

Activate QSPI interface

Triggering this task activates the external flash memory interface and initiates communication with the external memory. The READY event is generated when the activation has been completed.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_ACTIVATE			Activate QSPI interface
					Triggering this task activates the external flash memory interface and initiates communication with the external memory. The READY event is generated when the activation has been completed.
			Trigger	1	Trigger task

7.25.11.2 TASKS_READSTART

Address offset: 0x004

Start transfer from external flash memory to internal RAM

The READY event will be generated when transfer is complete.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_READSTART			Start transfer from external flash memory to internal RAM
					The READY event will be generated when transfer is complete.
			Trigger	1	Trigger task

7.25.11.3 TASKS_WRITESTART

Address offset: 0x008

Start transfer from internal RAM to external flash memory

The READY event will be generated when transfer is complete.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_WRITESTART			Start transfer from internal RAM to external flash memory The READY event will be generated when transfer is complete.
			Trigger	1	Trigger task

7.25.11.4 TASKS_ERASESTART

Address offset: 0x00C

Start external flash memory erase operation

The READY event will be generated when the erase operation has been started. The generation of the READY event does not imply that the erase operation is completed.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_ERASESTART			Start external flash memory erase operation The READY event will be generated when the erase operation has been started. The generation of the READY event does not imply that the erase operation is completed.
			Trigger	1	Trigger task

7.25.11.5 TASKS_DEACTIVATE

Address offset: 0x010

Deactivate QSPI interface

This task might be needed to optimize current consumption in case there are any added current consumption when QSPI interface is activated, but idle.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_DEACTIVATE			Deactivate QSPI interface This task might be needed to optimize current consumption in case there are any added current consumption when QSPI interface is activated, but idle.
			Trigger	1	Trigger task

7.25.11.6 SUBSCRIBE_ACTIVATE

Address offset: 0x080

Subscribe configuration for task ACTIVATE

Triggering this task activates the external flash memory interface and initiates communication with the external memory. The READY event is generated when the activation has been completed.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task ACTIVATE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.25.11.7 SUBSCRIBE_READSTART

Address offset: 0x084

Subscribe configuration for task READSTART

The READY event will be generated when transfer is complete.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task READSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.25.11.8 SUBSCRIBE_WRITESTART

Address offset: 0x088

Subscribe configuration for task WRITESTART

The READY event will be generated when transfer is complete.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task WRITESTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.25.11.9 SUBSCRIBE_ERASESTART

Address offset: 0x08C

Subscribe configuration for task ERASESTART

The READY event will be generated when the erase operation has been started. The generation of the READY event does not imply that the erase operation is completed.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task ERASESTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.25.11.10 SUBSCRIBE_DEACTIVATE

Address offset: 0x090

Subscribe configuration for task DEACTIVATE

This task might be needed to optimize current consumption in case there are any added current consumption when QSPI interface is activated, but idle.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task DEACTIVATE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.25.11.11 EVENTS_READY

Address offset: 0x100

QSPI peripheral is ready. This event will be generated as a response to all QSPI tasks except DEACTIVATE.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_READY			QSPI peripheral is ready. This event will be generated as a response to all QSPI tasks except DEACTIVATE.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.25.11.12 PUBLISH_READY

Address offset: 0x180

Publish configuration for event READY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.25.11.13 INTEN

Address offset: 0x300

Enable or disable interrupt

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

7.25.11.14 INTENSET

Address offset: 0x304

Enable interrupt

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

7.25.11.15 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.25.11.16 ENABLE

Address offset: 0x500

Enable QSPI peripheral and acquire the pins selected in PSELn registers

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable QSPI
			Disabled	0	Disable QSPI
			Enabled	1	Enable QSPI

7.25.11.17 READ.SRC

Address offset: 0x504

Flash memory source address

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SRC			Word-aligned flash memory source address.

7.25.11.18 READ.DST

Address offset: 0x508

RAM destination address

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

ID	R/W	Field	Value ID	Value	Description
A	RW	DST			Word-aligned RAM destination address.

7.25.11.19 READ.CNT

Address offset: 0x50C

Read transfer length

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CNT			Read transfer length in number of bytes. The length must be a multiple of 4 bytes.

7.25.11.20 WRITE.DST

Address offset: 0x510

Flash destination address

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	DST			Word-aligned flash destination address.

7.25.11.21 WRITE.SRC

Address offset: 0x514

RAM source address

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	SRC			Word-aligned RAM source address.

7.25.11.22 WRITE.CNT

Address offset: 0x518

Write transfer length

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CNT			Write transfer length in number of bytes. The length must be a multiple of 4 bytes.

7.25.11.23 ERASE.PTR

Address offset: 0x51C

Start address of flash block to be erased

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	PTR			Word-aligned start address of block to be erased.

7.25.11.24 ERASE.LEN

Address offset: 0x520

Size of block to be erased.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LEN			LEN
			4KB	0	Erase 4 kB block (flash command 0x20)
			64KB	1	Erase 64 kB block (flash command 0xD8)
			All	2	Erase all (flash command 0xC7)

7.25.11.25 PSEL.SCK

Address offset: 0x524

Pin select for serial clock SCK

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.25.11.26 PSEL.CSN

Address offset: 0x528

Pin select for chip select signal CSN.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.25.11.27 PSEL.IO0

Address offset: 0x530

Pin select for serial data MOSI/IO0.

Serial data output (MOSI) during single mode, or serial data IO0 during quad mode

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

7.25.11.28 PSEL.IO1

Address offset: 0x534

Pin select for serial data MISO/IO1.

Serial data input (MISO) during single mode, or serial data IO1 during quad mode

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

7.25.11.29 PSEL.IO2

Address offset: 0x538

Pin select for serial data WP/IO2.

In single mode, this pin can control Write protect (WP, active 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				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.25.11.30 PSEL.IO3

Address offset: 0x53C

Pin select for serial data HOLD/IO3.

In single mode, this pin can can pause the device (HOLD, active 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				C B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.25.11.31 XIPOFFSET

Address offset: 0x540

Address offset into the external memory for Execute in Place operation.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	XIPOFFSET			Address offset into the external memory for Execute in Place operation. Value must be a multiple of 4.

7.25.11.32 IFCONFIG0

Address offset: 0x544

Interface configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G D C B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	READOC			Configure number of data lines and opcode used for reading.
			FASTREAD	0	Single data line SPI. FAST_READ (opcode 0x0B).
			READ2O	1	Dual data line SPI. READ2O (opcode 0x3B).
			READ2IO	2	Dual data line SPI. READ2IO (opcode 0xBB).
			READ4O	3	Quad data line SPI. READ4O (opcode 0x6B).
			READ4IO	4	Quad data line SPI. READ4IO (opcode 0xEB).
B	RW	WRITEOC			Configure number of data lines and opcode used for writing.
			PP	0	Single data line SPI. PP (opcode 0x02).
			PP2O	1	Dual data line SPI. PP2O (opcode 0xA2).
			PP4O	2	Quad data line SPI. PP4O (opcode 0x32).
			PP4IO	3	Quad data line SPI. PP4IO (opcode 0x38).
C	RW	ADDRMODE			Addressing mode.
			24BIT	0	24-bit addressing.
			32BIT	1	32-bit addressing.
D	RW	DPMENABLE			Enable deep power-down mode (DPM) feature.
			Disable	0	Disable DPM feature.
			Enable	1	Enable DPM feature.
G	RW	PPSIZE			Page size for commands PP, PP2O, PP4O and PP4IO.
			256Bytes	0	256 bytes.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G D C B B B A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			512Bytes	1	512 bytes.

7.25.11.33 XIPEN

Address offset: 0x54C

Enable Execute in Place operation.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	XIPEN			Enable XIP AHB Slave interface and access to XIP memory range When disabled, access to external memory is only available through EasyDMA and custom instructions. Access to disabled XIP interface will cause a Bus Error, and the value read is all zeros.
			Disable	0	Disable XIP interface
			Enable	1	Enable XIP interface

7.25.11.34 XIP_ENC.KEY0

Address offset: 0x560

Bits 31:0 of XIP AES KEY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY0			Bits 31:0 of XIP AES KEY

7.25.11.35 XIP_ENC.KEY1

Address offset: 0x564

Bits 63:32 of XIP AES KEY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY1			Bits 63:32 of XIP AES KEY

7.25.11.36 XIP_ENC.KEY2

Address offset: 0x568

Bits 95:64 of XIP AES KEY

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY2			Bits 95:64 of XIP AES KEY

7.25.11.37 XIP_ENC.KEY3

Address offset: 0x56C

Bits 127:96 of XIP AES KEY

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY3			Bits 127:96 of XIP AES KEY

7.25.11.38 XIP_ENC.NONCE0

Address offset: 0x570

Bits 31:0 of XIP NONCE

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	NONCE0			Bits 31:0 of XIP NONCE

7.25.11.39 XIP_ENC.NONCE1

Address offset: 0x574

Bits 63:32 of XIP NONCE

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	NONCE1			Bits 63:32 of XIP NONCE

7.25.11.40 XIP_ENC.NONCE2

Address offset: 0x578

Bits 95:64 of XIP NONCE

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	NONCE2			Bits 95:64 of XIP NONCE

7.25.11.41 XIP_ENC.ENABLE

Address offset: 0x57C

Enable stream cipher for XIP

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable stream cipher for XIP
			Disabled	0	Disable stream cipher for QSPI XIP
			Enabled	1	Enable stream cipher for QSPI XIP

7.25.11.42 DMA_ENC.KEY0

Address offset: 0x580

Bits 31:0 of DMA AES KEY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY0			Bits 31:0 of DMA AES KEY

7.25.11.43 DMA_ENC.KEY1

Address offset: 0x584

Bits 63:32 of DMA AES KEY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY1			Bits 63:32 of DMA AES KEY

7.25.11.44 DMA_ENC.KEY2

Address offset: 0x588

Bits 95:64 of DMA AES KEY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	KEY2			Bits 95:64 of DMA AES KEY

7.25.11.45 DMA_ENC.KEY3

Address offset: 0x58C

Bits 127:96 of DMA AES KEY

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	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	R/W	Field	Value ID	Value	Description
A	W	KEY3			Bits 127:96 of DMA AES KEY

7.25.11.46 DMA_ENC.NONCE0

Address offset: 0x590

Bits 31:0 of DMA NONCE

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	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	R/W	Field	Value ID	Value	Description
A	W	NONCE0			Bits 31:0 of DMA NONCE

7.25.11.47 DMA_ENC.NONCE1

Address offset: 0x594

Bits 63:32 of DMA NONCE

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	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	R/W	Field	Value ID	Value	Description
A	W	NONCE1			Bits 63:32 of DMA NONCE

7.25.11.48 DMA_ENC.NONCE2

Address offset: 0x598

Bits 95:64 of DMA NONCE

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	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	R/W	Field	Value ID	Value	Description
A	W	NONCE2			Bits 95:64 of DMA NONCE

7.25.11.49 DMA_ENC.ENABLE

Address offset: 0x59C

Enable stream cipher for EasyDMA

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

7.25.11.50 IFCONFIG1

Address offset: 0x600

Interface configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G G G G E D A A A A A A A A
Reset 0x00040480				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	SCKDELAY		[0..255]	Minimum amount of time that the CSN pin must stay high before it can go low again. Value is specified in number of 16 MHz periods (62.5 ns).
D	RW	DPMEN			Enter/exit deep power-down mode (DPM) for external flash memory.
			Exit	0	Exit DPM.
			Enter	1	Enter DPM.
E	RW	SPIMODE			Select SPI mode.
			MODE0	0	Mode 0: Data are captured on the clock rising edge and data is output on a falling edge. Base level of clock is 0 (CPOL=0, CPHA=0).
			MODE3	1	Mode 3: Data are captured on the clock rising edge and data is output on a falling edge. Base level of clock is 1 (CPOL=1, CPHA=1).
G	RW	SCKFREQ		[0..15]	SCK frequency is derived from PCLK192M with SCK frequency = PCLK192M / (2*(SCKFREQ + 1)).

7.25.11.51 STATUS

Address offset: 0x604

Status register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F F F F F F F F D C
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
C	R	DPM			Deep power-down mode (DPM) status of external flash.
			Disabled	0	External flash is not in DPM.
			Enabled	1	External flash is in DPM.
D	R	READY			Ready status.
			READY	1	QSPI peripheral is ready. It is allowed to trigger new tasks, writing custom instructions or enter/exit DPM.
			BUSY	0	QSPI peripheral is busy. It is not allowed to trigger any new tasks, writing custom instructions or enter/exit DPM.
F	R	SREG			Value of external flash device Status Register. When the external flash has two bytes status register this field includes the value of the low byte.

7.25.11.52 DPMDUR

Address offset: 0x614

Set the duration required to enter/exit deep power-down mode (DPM).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B B B B B B B B B A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	ENTER		[0..0xFFFF]	Duration needed by external flash to enter DPM. Duration is given as ENTER * 256 * 62.5 ns.
B	RW	EXIT		[0..0xFFFF]	Duration needed by external flash to exit DPM. Duration is given as EXIT * 256 * 62.5 ns.

7.25.11.53 ADDRCONF

Address offset: 0x624

Extended address configuration.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D D C C C C C C C B B B B B B B A A A A A A A A
Reset 0x000000B7				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	OPCODE		[0xFF..0]	Opcode that enters the 32-bit addressing mode.
B	RW	BYTE0		[0xFF..0]	Byte 0 following opcode.
C	RW	BYTE1		[0xFF..0]	Byte 1 following byte 0.
D	RW	MODE			Extended addressing mode.
			NoInstr	0	Do not send any instruction.
			Opcode	1	Send opcode.
			OpByte0	2	Send opcode, BYTE0.
			All	3	Send opcode, BYTE0, BYTE1.
E	RW	WIPWAIT			Wait for write complete before sending command.
			Disable	0	No wait.
			Enable	1	Wait.
F	RW	WREN			Send WREN (write enable opcode 0x06) before instruction.
			Disable	0	Do not send WREN.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x000000B7				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
			Enable	1	Send WREN.

7.25.11.54 CINSTRCONF

Address offset: 0x634

Custom instruction configuration register.

A new custom instruction is sent every time this register is written. The READY event will be generated when the custom instruction has been sent.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B B B B A A A A A A A A
Reset 0x00002000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	OPCODE		[0..255]	Opcode of Custom instruction.
B	RW	LENGTH			Length of custom instruction in number of bytes.
			1B	1	Send opcode only.
			2B	2	Send opcode, CINSTRDAT0.BYTE0.
			3B	3	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE1.
			4B	4	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE2.
			5B	5	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT0.BYTE3.
			6B	6	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE4.
			7B	7	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE5.
			8B	8	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE6.
			9B	9	Send opcode, CINSTRDAT0.BYTE0 -> CINSTRDAT1.BYTE7.
C	RW	LIO2		[0..1]	Level of the IO2 pin (if connected) during transmission of custom instruction.
D	RW	LIO3		[0..1]	Level of the IO3 pin (if connected) during transmission of custom instruction.
E	RW	WIPWAIT			Wait for write complete before sending command.
			Disable	0	No wait.
			Enable	1	Wait.
F	RW	WREN			Send WREN (write enable opcode 0x06) before instruction.
			Disable	0	Do not send WREN.
			Enable	1	Send WREN.
G	RW	LFEN			Enable Long frame mode. When enabled, a custom instruction transaction has to be ended by writing the LFSTOP field.
			Disable	0	Long frame mode disabled
			Enable	1	Long frame mode enabled
H	RW	LFSTOP			Stop (finalize) long frame transaction
			Stop	1	Stop

7.25.11.55 CINSTRDAT0

Address offset: 0x638

Custom instruction data register 0.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	BYTE0		[0..0xFF]	Data byte 0
B	RW	BYTE1		[0..0xFF]	Data byte 1
C	RW	BYTE2		[0..0xFF]	Data byte 2
D	RW	BYTE3		[0..0xFF]	Data byte 3

7.25.11.56 CINSTRDAT1

Address offset: 0x63C

Custom instruction data register 1.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	BYTE4		[0..0xFF]	Data byte 4
B	RW	BYTE5		[0..0xFF]	Data byte 5
C	RW	BYTE6		[0..0xFF]	Data byte 6
D	RW	BYTE7		[0..0xFF]	Data byte 7

7.25.11.57 IFTIMING

Address offset: 0x640

SPI interface timing.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C C
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
C	RW	RXDELAY		[7..0]	Timing related to sampling of the input serial data. The value of RXDELAY specifies the number of prescaled 192 MHz cycles delay from the the rising edge of the SPI Clock (SCK) until the input serial data is sampled. For example, if RXDELAY is set to 0, the input serial data is sampled on the rising edge of SCK.

7.25.12 Electrical specification

7.25.12.1 Timing specification

Symbol	Description	Min.	Typ.	Max.	Units
$F_{QSPI,CLK}$	SCK frequency			96	MHz
$DC_{QSPI,CLK}$	SCK duty cycle	..	..	..	%
$F_{QSPI,XIP,16}$	XIP fetch frequency for 16 bit instructions			24	MHz
$F_{QSPI,XIP,32}$	XIP fetch frequency for 32 bit instructions			12	MHz
t_{MOH}	External memory output hold time	..	..	..	ns

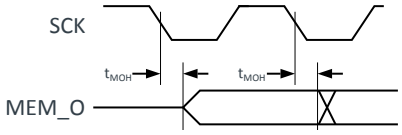

Figure 139: QSPI memory timing diagram

7.26 RADIO — 2.4 GHz radio

The 2.4 GHz radio transceiver is compatible with multiple radio standards such as Bluetooth Low Energy, IEEE 802.15.4, and Nordic's proprietary modes.

The main features of RADIO are:

- Multidomain 2.4 GHz radio transceiver, with
  - Bluetooth Low Energy 1 Mbps and 2 Mbps modes
  - Bluetooth Low Energy Long Range (125 kbps and 500 kbps) modes
  - Angle of Arrival (AoA) and Angle of Departure (AoD) direction finding using Bluetooth Low Energy
  - IEEE 802.15.4 250 kbps mode
  - 1 Mbps and 2 Mbps Nordic proprietary modes
- Best in class link budget and low power operation
- Efficient data interface with EasyDMA support
- Automatic address filtering and pattern matching

EasyDMA, in combination with an automated packet assembler, packet disassembler, automated CRC generator and CRC checker, makes it easy to configure and use RADIO. See the following figure for details.

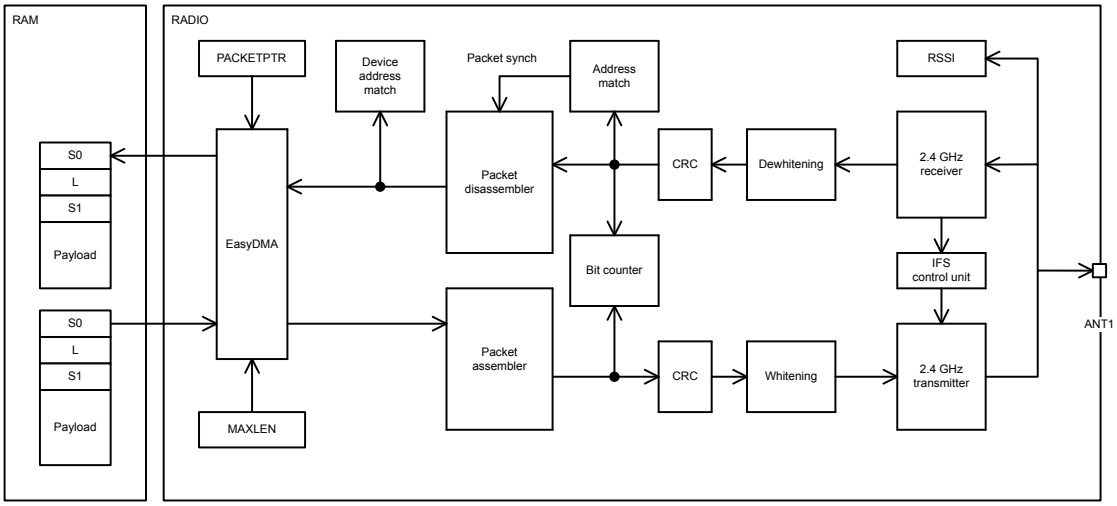

Figure 140: RADIO block diagram

RADIO includes a device address match unit and an interframe spacing control unit that can be utilized to simplify address whitelisting and interframe spacing respectively in Bluetooth Low Energy and similar applications.

RADIO also includes a received signal strength indicator (RSSI) and a bit counter. The bit counter generates events when a preconfigured number of bits are sent or received by RADIO.

7.26.1 Packet configuration

A RADIO packet contains the fields PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD, and CRC. For Long Range (125 kbps and 500 kbps) Bluetooth Low Energy modes, fields CI, TERM1, and TERM2 are also included.

The content of a RADIO packet is illustrated in the following figures. RADIO sends the fields in the packet according to the sequence shown in the figures, starting on the left.

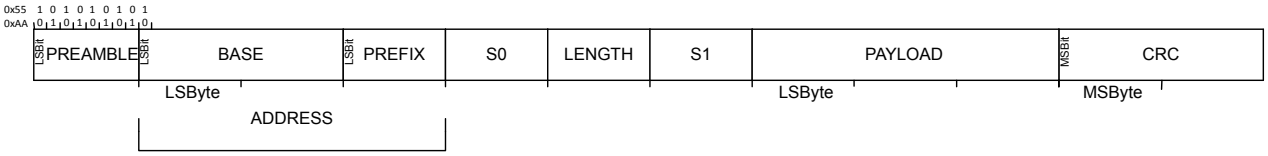

Figure 141: On-air packet layout

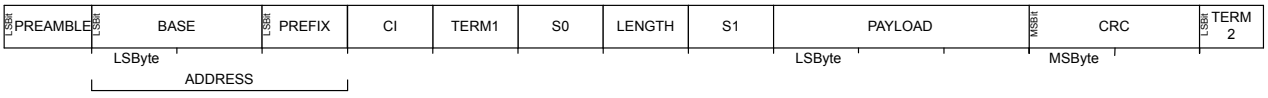

Figure 142: On-air packet layout for Long Range (125 kbps and 500 kbps) Bluetooth Low Energy modes

Not shown in the figures is the static payload add-on (the length of which is defined in PCNF1.STATLEN, and which is 0 bytes in a standard BLE packet). The static payload add-on is sent between PAYLOAD and CRC fields. RADIO sends the different fields in the packet in the order they are illustrated above, from left to right.

PREAMBLE is sent with least significant bit first on air. The size of the PREAMBLE depends on the mode selected in the MODE register:

- The PREAMBLE is one byte for MODE = Ble_1Mbit as well as all Nordic proprietary operating modes (MODE = Nrf_1Mbit and MODE = Nrf_2Mbit), and PCNF0.PLEN has to be set accordingly. If the first bit of the ADDRESS is 0, the preamble will be set to 0xAA. Otherwise the PREAMBLE will be set to 0x55.
- For MODE = Ble_2Mbit, the PREAMBLE must be set to 2 bytes through PCNF0.PLEN. If the first bit of the ADDRESS is 0, the preamble will be set to 0xAAAA. Otherwise the PREAMBLE will be set to 0x5555.
- For MODE = Ble_LR125Kbit and MODE = Ble_LR500Kbit, the PREAMBLE is 10 repetitions of 0x3C.
- For MODE = Ieee802154_250Kbit, the PREAMBLE is 4 bytes and set to all zeros.

Radio packets are stored in memory inside instances of a RADIO packet data structure as illustrated below. The PREAMBLE, ADDRESS, CI, TERM1, TERM2, and CRC fields are omitted in this data structure. Fields S0, LENGTH, and S1 are optional.

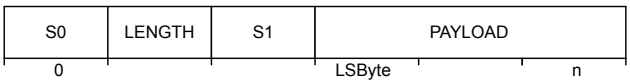

Figure 143: Representation of a RADIO packet in RAM

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. The CRC field is always transmitted and received most significant bit first. The endianness, i.e. the order in which the bits are sent and received, of the S0, LENGTH, S1, and PAYLOAD fields can be configured via PCNF1.ENDIAN.

The sizes of the S0, LENGTH, and S1 fields can be individually configured via S0LEN, LFLEN, and S1LEN in PCNF0 respectively. If any of these fields are configured to be less than 8 bits, the least significant bits of the fields are used.

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

Independent of the configuration of PCNF1.MAXLEN, the combined length of S0, LENGTH, S1, and PAYLOAD cannot exceed 258 bytes.

7.26.2 Address configuration

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

The size of the base address field is configurable via PCNF1.BALEN. The base address is truncated from the least significant byte if the PCNF1.BALEN is less than 4. See Definition of logical addresses on page 424.

The on-air addresses are defined in the BASE0/BASE1 and PREFIX0/PREFIX1 registers. It is only when writing these registers that the user must relate to the 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 the following table.

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

Table 121: Definition of logical addresses

7.26.3 Data whitening

RADIO can do packet whitening and de-whitening which is enabled in PCNF1.WHITEEN. When enabled, whitening and de-whitening will be handled by 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. The linear feedback shift register is initialized via DATAWHITEIV. See the following figure.

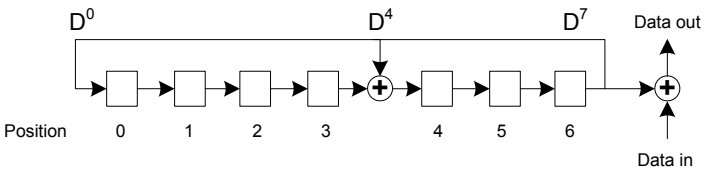

Figure 144: Data whitening and de-whitening

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

Including the address field in CRC check (CRCCNF.SKIPADDR=Include) is not supported for whitened packets.

7.26.4 CRC

The CRC generator in RADIO calculates the CRC over the whole packet excluding the preamble.

If useful, the address field can be excluded from the CRC calculation as well. See the CRCCNF register for more information.

The CRC polynomial is configurable as illustrated in the following figure, where bit `0` in the CRCPOLY register corresponds to X^0 and bit `1` corresponds to X^1 etc. See CRCPOLY on page 482 for more information.

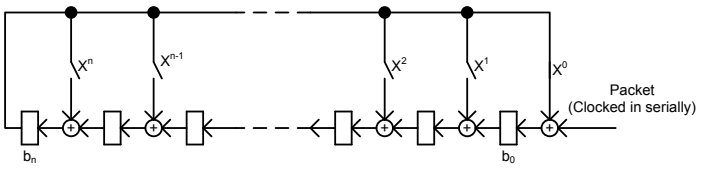

Figure 145: CRC generation of an n bit CRC

The figure shows that 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. After the whole packet has been clocked through the CRC generator, b_0 through b_n will hold the resulting CRC. This value will be used by RADIO during both transmission and reception. Latches b_0 through b_n are not available to be read by the CPU at any time. However, a received CRC can be read by the CPU via the RXCRC register.

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

Once the entire packet, including the CRC, has been received and no errors were detected, RADIO generates a CRCOK event. If CRC errors were detected, a CRCERROR event is generated.

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

7.26.5 Radio states

Tasks and events are used to control the operating state of RADIO.

RADIO can enter the states described in the following table.

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

Table 122: RADIO state diagram

A state diagram showing an overview of RADIO is shown in the following figure.

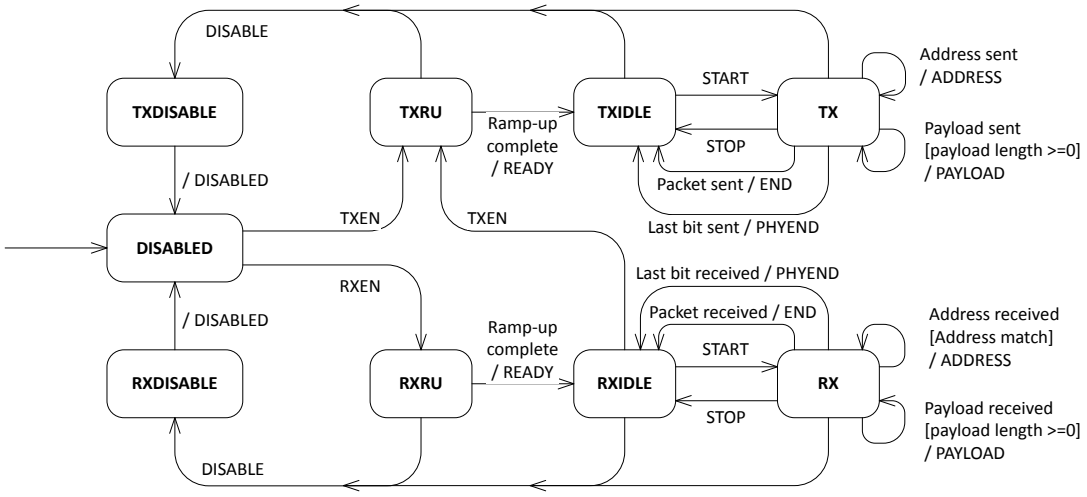

Figure 146: Radio states

This figure shows how the tasks and events relate to RADIO's operation. 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 behavior. The PAYLOAD event is always generated even if the payload is zero.

The END to START shortcut should not be used with IEEE 802.15.4 250 kbps mode. Use the PHYEND to START shortcut instead.

The END to START shortcut should not be used with Long Range (125 kbps and 500 kbps) Bluetooth Low Energy modes. Use the PHYEND to START shortcut instead.

7.26.6 Transmit sequence

Before RADIO can transmit a packet, it must first ramp-up in TX mode. See TXRU in Radio states on page 426 and Transmit sequence on page 427. A TXRU ramp-up sequence is initiated when the TXEN task is triggered. After RADIO has successfully ramped up it will generate the READY event indicating that a packet transmission can be initiated. A packet transmission is initiated by triggering the START task. The START task can first be triggered after RADIO has entered into the TXIDLE state.

The following figure illustrates a single packet transmission where the CPU manually triggers the different tasks needed to control the flow of 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 the following figure, 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.

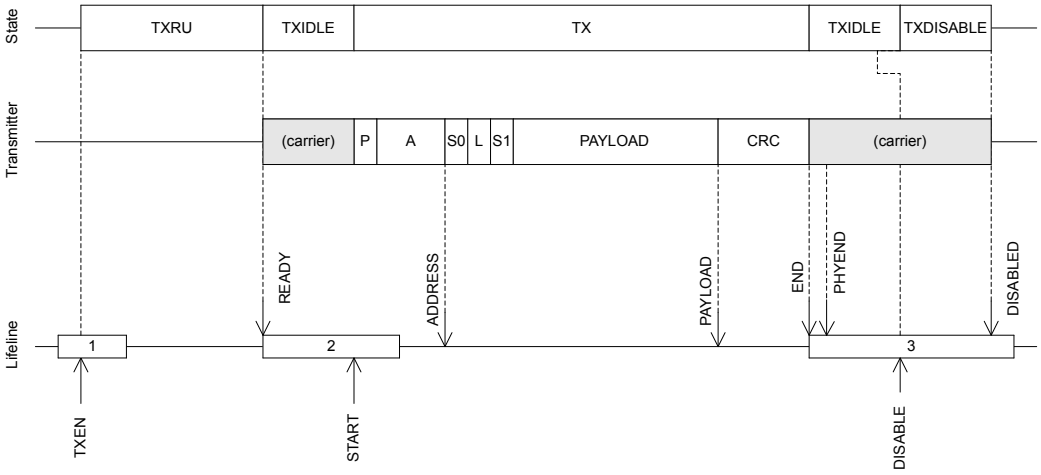

Figure 147: Transmit sequence

The following figure shows a slightly modified version of the transmit sequence where RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.

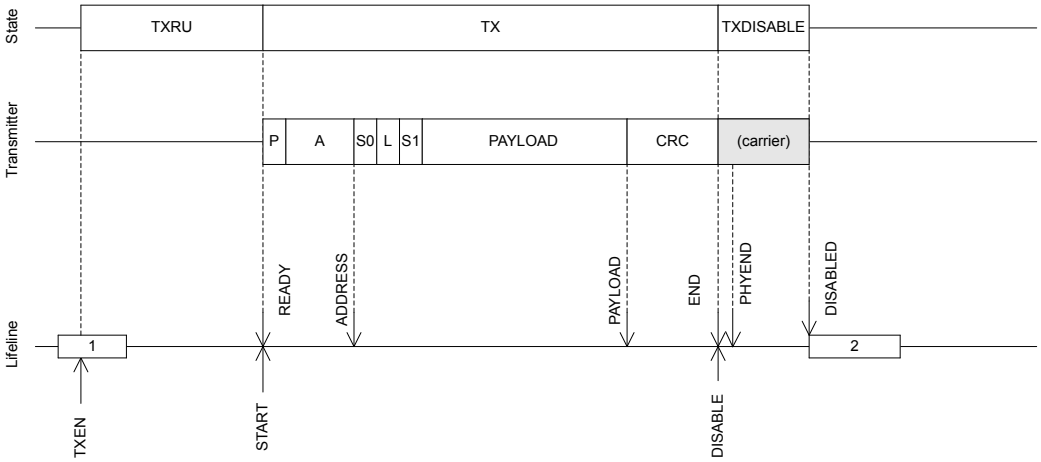

Figure 148: Transmit sequence using shortcuts to avoid delays

RADIO is able to send multiple packets one after the other without having to disable and re-enable RADIO between packets, as illustrated in the following figure.

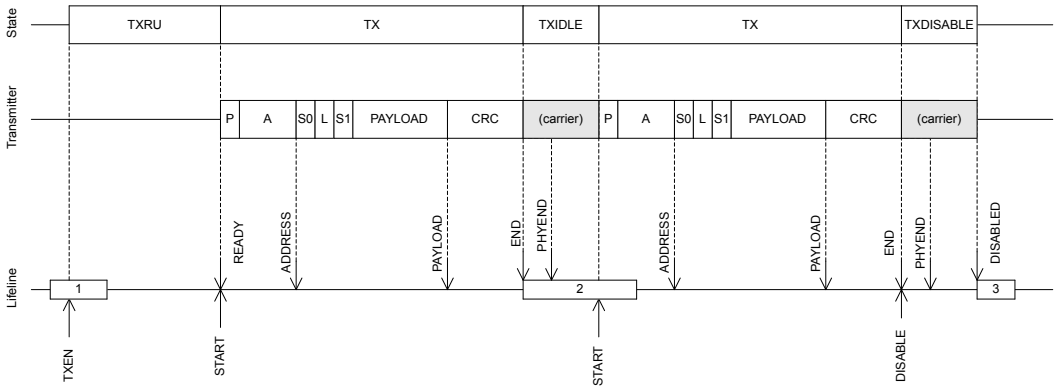

Figure 149: Transmission of multiple packets

7.26.7 Receive sequence

Before RADIO is able to receive a packet, it must first ramp up in RX mode. See RXRU in Radio states on page 426 and Receive sequence on page 428 for more information.

An RXRU ramp up sequence is initiated when the RXEN task is triggered. After RADIO has successfully ramped up it will generate the READY event indicating that a packet reception can be initiated. A packet reception is initiated by triggering the START task. As illustrated in Radio states on page 426, the START task can first be triggered after RADIO has entered into the RXIDLE state.

The following figure shows a single packet reception where the CPU manually triggers the different tasks needed to control the flow of 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. RADIO will be listening and possibly receiving undefined data, represented with an 'X', from START and until a packet with valid preamble (P) is received.

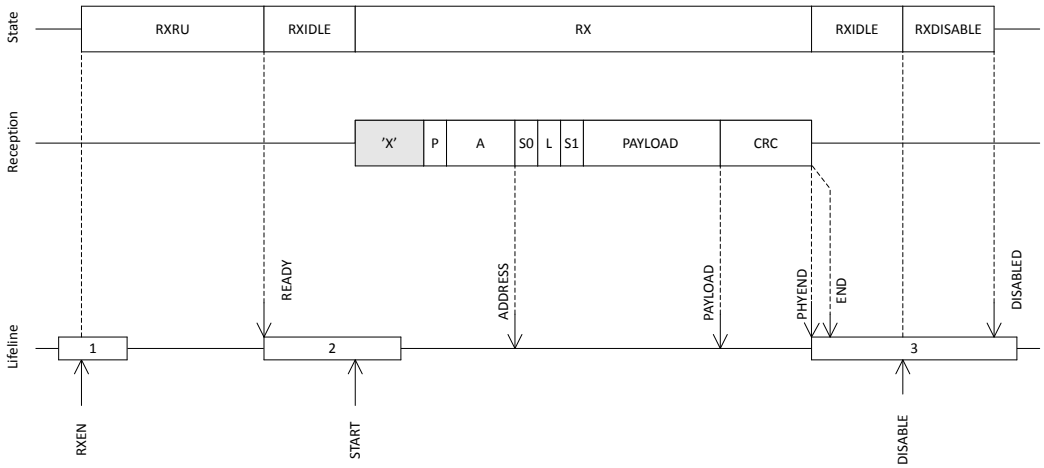

Figure 150: Receive sequence

The following figure shows a modified version of the receive sequence, where RADIO is configured to use shortcuts between READY and START, and between END and DISABLE, which means that no delay is introduced.

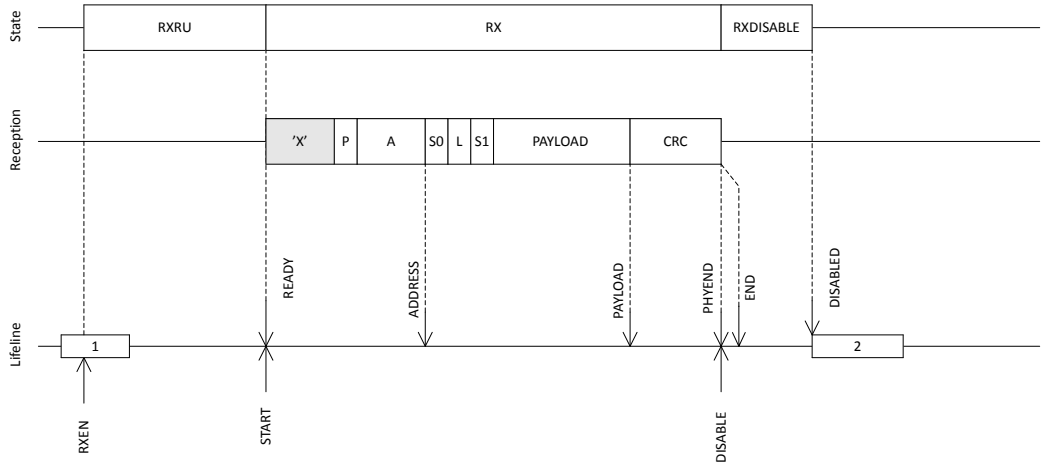

Figure 151: Receive sequence using shortcuts to avoid delays

RADIO can receive consecutive packets without having to disable and re-enable RADIO between packets, as illustrated in the following figure.

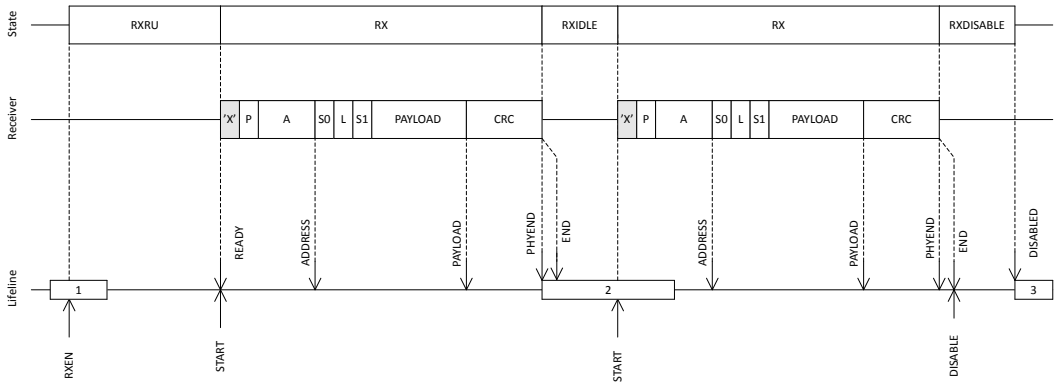

Figure 152: Reception of multiple packets

7.26.8 Received signal strength indicator (RSSI)

RADIO implements a mechanism for measuring the power in the received signal. This feature is called received signal strength indicator (RSSI).

The RSSI is measured continuously and the value filtered using a single-pole IIR filter. After a signal level change, the RSSI will settle after approximately $RSSI_{SETTLE}$.

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

The sample period of the RSSI is defined by $RSSI_{PERIOD}$. The RSSISAMPLE will hold the filtered received signal strength after this sample period.

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

7.26.9 Interframe spacing (IFS)

Interframe spacing (IFS) is defined as the time, in microseconds, between two consecutive packets, starting from when the end of the last bit of the previous packet is received, to the beginning of the first bit of the subsequent packet that is transmitted.

RADIO can enforce this interval, as specified in the TIFS register, as long as the TIFS register is not specified to be shorter than RADIO's turnaround time (i.e. the time needed to switch off the receiver, and then switch the transmitter back on). The TIFS register can be written any time before the last bit on air is received.

This timing is illustrated in the following figure.

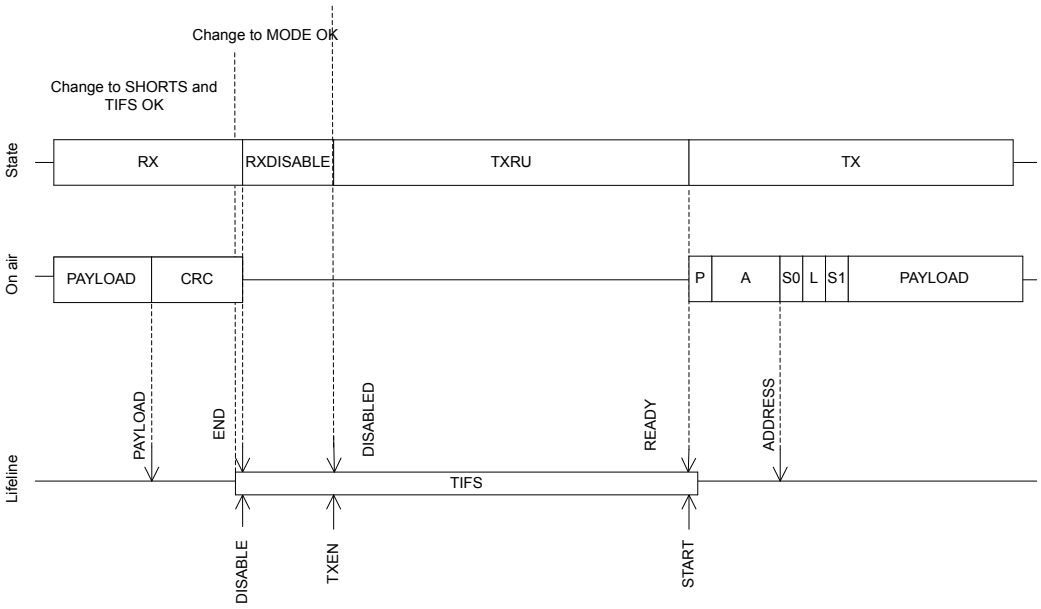

Figure 153: IFS timing detail

The TIFS duration starts after the last bit on air (just before the END event), and elapses with the first bit being transmitted on air (just after READY event).

TIFS is only enforced if the shortcuts END to DISABLE and DISABLED to TXEN or END to DISABLE and DISABLED to RXEN are enabled.

TIFS is qualified for use in IEEE 802.15.4 250kbps mode, Bluetooth Low Energy Long Range (125 kbps and 500 kbps) modes, and Bluetooth Low Energy 1 Mbps and 2 Mbps modes, using the default ramp-up mode.

SHORTS and TIFS registers are not double-buffered, and can be updated at any point before the last bit on air is received. The MODE register is double-buffered and sampled at the TXEN or RXEN task.

7.26.10 Device address match

The device address match feature is tailored for address whitelisting in Bluetooth Low Energy and similar implementations.

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

The device address match unit assumes that the first 48 bits of the payload are 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.

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.

7.26.11 Bit counter

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 RADIO and 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. After a BCMATCH event, the CPU can reconfigure the BCC value for new BCMATCH events within the same packet.

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

The bit counter will stop and reset on either the BCSTOP, STOP, or DISABLE task, or the END event.

The following figure shows 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.

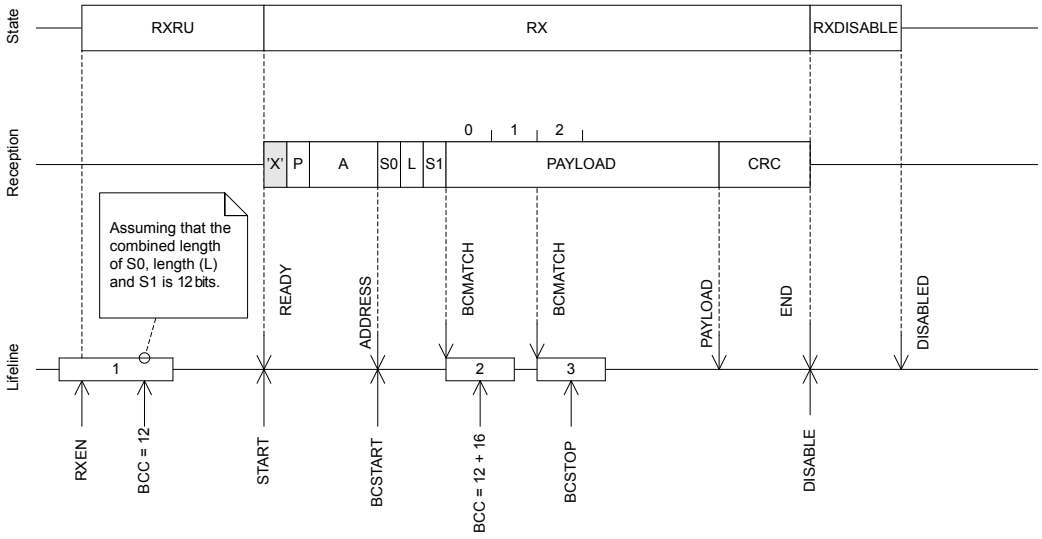

Figure 154: Bit counter example

7.26.12 Direction finding

RADIO implements the Angle of Arrival (AoA) and Angle of Departure (AoD) Bluetooth Low Energy feature, which can be used to determine the direction of a peer device. The feature is available for the Bluetooth Low Energy 1 and 2 Mbps modes.

When using this feature, the transmitter sends a packet with a constant tone extension (CTE) appended to the packet, after the CRC. During the CTE, the receiver can take IQ samples of the incoming signal.

An antenna array is employed at the transmitter (AoD) or at the receiver (AoA). The AoD transmitter, or AoA receiver, switches between the antennas, in order to collect IQ samples from the different antenna pairs. The IQ samples can be used to calculate the relative path lengths between the antenna pairs, which can be used to estimate the direction of the transmitter.

7.26.12.1 CTE format

The CTE is from 16 µs to 160 µs and consists of an unwhitened sequence of `1`s, equivalent to a continuous tone nominally offset from the carrier by +250 kHz for the Bluetooth Low Energy 1 Mbps PHY and +500 kHz for the 2 Mbps Bluetooth Low Energy PHY. The format of the CTE, when switching and/or sampling, is shown in the following figure.

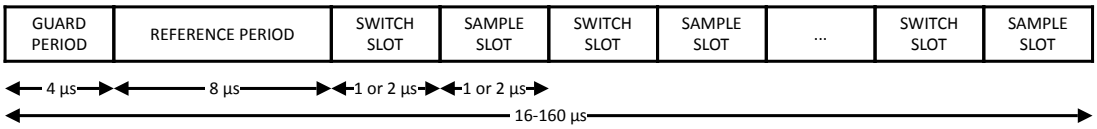

Figure 155: Constant tone extension (CTE) structure

Antenna switching is performed during switch slots and the guard period. The AoA/AoD feature requires that one IQ sample is taken for each microsecond within the reference period, and once for each sample slot. Oversampling is possible by changing the sample spacing as described in IQ sampling on page 435. The switch slot and sample slot durations are either 1 or 2 µs, but must be equal. The format of the CTE and switching and sampling procedures may be configured prior to, or during, packet transmission and reception. Alternatively, during packet reception, these operations can be configured by reading specific fields of the packet contents.

7.26.12.2 Mode

Depending on the DFEMODE, the device performs the procedures shown in the following table.

		DFEMODE			
		AOA		AOD	
		TX	RX	TX	RX
AoA/AoD Procedure	Generating and transmitting CTE	x		x	
	Receiving, interpreting, and sampling CTE		x		x
	Antenna switching		x	x	

Table 123: AoA/AoD Procedures performed as a function of DFEMODE and TX/RX mode

7.26.12.3 Inline configuration

When inline configuration is enabled during RX, further configuration of the AoA/AoD procedures is performed based on the values of the CP bit and the CTEInfo octet within the packet. This is enabled by setting CTEINLINECONF.CTEINLINECTRLEN. The CTEInfo octet is present only if the CP bit is set. The position of the CP bit and CTEInfo octet depends on whether the packet has a *Data Channel PDU* (CTEINLINECONF.CTEINFOINS1=InS1), or an *Advertising Channel PDU* (CTEINLINECONF.CTEINFOINS1=NotInS1).

Data channel PDU

For Data Channel PDUs, PCNF0.S0LEN must be 1 byte, and PCNF0.LFLEN must be 8 bits. To determine if S1 is present, the registers CTEINLINECONF.S0MASK and CTEINLINECONF.S0CONF forms a bitwise mask-and-test for the S0 field. If the bitwise AND between S0 and S0MASK equals S0CONF, then S1 is determined to be present. When present, the value of PCNF0.S1LEN will be ignored, as this is decided by the CP bit in the the following figure.

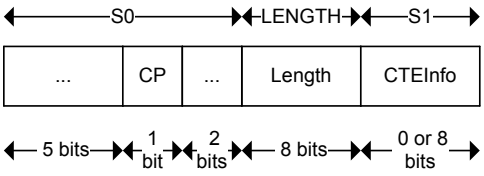

Figure 156: Data channel PDU header

When encrypting and decrypting Bluetooth Low Energy packets using the CCM peripheral, it is also required to set PCNF0.S1INCL=1. The CCM mode must be configured to use an 8-bit length field. The value of the CP bit is included in the calculation of the MIC, while the S1 field is ignored by the CCM calculation.

Advertising channel PDU

For advertising channel PDUs, the CTEInfo Flag replaces the CP bit. The CTEInfo Flag is within the extended header flag field in some of the advertising PDUs that employ the common extended advertising payload format (i.e. AUX_SYNC_IND, AUX_CHAIN_IND). The format of such packets is shown in the following figure.

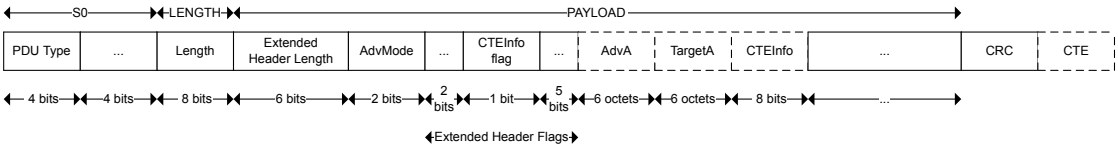

Figure 157: Advertising channel PDU header

The CTEINLINECONF.S0CONF and CTEINLINECONF.S0MASK fields can be configured to accept only certain advertising PDU Types. If the extended header length is non-zero, the CTEInfo extended header flag is checked to determine whether CTEInfo is present. If a bit before the CTEInfo flag within the extended header flags is set, then the CTEInfo position is postponed 6 octets.

CTEInfo parsing

The CTEInfo field is shown in the following figure.

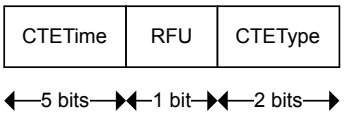

Figure 158: CTEInfo field

The CTETIME field defines the length of the CTE in 8 µs units. The valid upper bound of values can be adjusted using CTEINLINECONF.CTETIMEVALIDRANGE, including allowing use of the RFU bit within this field. If the CTETIME field is an invalid value of either 0 or 1, the CTE is assumed to be the minimum valid length of 16 µs. The slot duration is determined by the CTEType field. In RX mode this determines whether the sample spacing as defined in CTEINLINECONF.CTEINLINERXMODE1US or CTEINLINECONF.CTEINLINERXMODE2US is used.

CTEType	Description	TX switch spacing	RX sample spacing during reference period	Sample spacing RX during reference period
0	AoA, no switching	-	TSAMPLESPACING1	TSAMPLESPACING2
1	AoD, 1 µs slots	2 µs	TSAMPLESPACING1	CTEINLINERXMODE1US
2	AoD, 2 µs slots	4 µs	TSAMPLESPACING1	CTEINLINERXMODE2US
3	Reserved for future use			

Table 124: Switching and sampling spacing based on CTEType

7.26.12.4 Manual configuration

If CTEINLINECONF.CTEINLINECTRLEN is not set, then the packet is not parsed to determine the CTE parameters and the antenna switching and sampling is controlled by other registers (see Antenna switching on page 434). The length of the CTE is given in 8 µs units by DFECTRL1.NUMBEROF8US. The start of the antenna switching and/or sampling (denoted as an AoA/AoD procedure), can be configured to start at some trigger with an additional offset. Using DFECTRL1.DFEINEXTENSION, the trigger can be configured to be the end of the CRC, or alternatively, the ADDRESS event. The additional offset for antenna switching is configured using DFECTRL2.TSWITCHOFFSET. Similarly, the additional offset for antenna sampling is configured using DFECTRL2.TSAMPLEOFFSET.

7.26.12.5 Receive- and transmit sequences

The addition of the CTE to the transmitted packet is illustrated in the following figure.

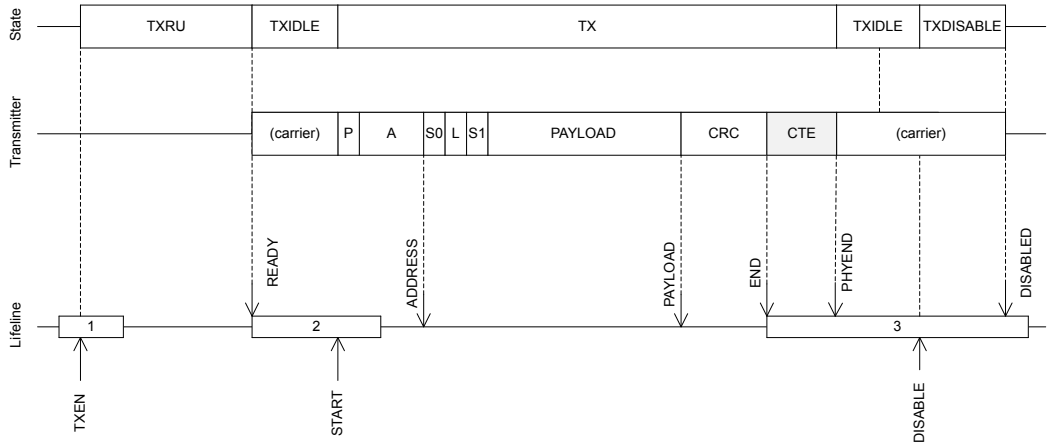

Figure 159: Transmit sequence with DFE

The presence of CTE within a received packet is signalled by the CTEPRESENT event illustrated in the following figure.

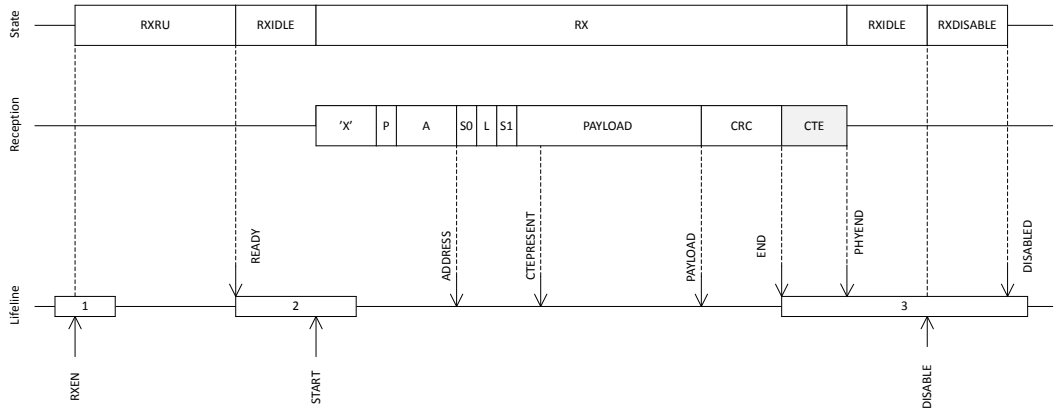

Figure 160: Receive sequence with DFE

7.26.12.6 Antenna switching

RADIO can control up to 8 GPIO pins in order to control external antenna switches used in direction finding.

Pin configuration

The eight antenna selection signals are mapped to physical pins according to the pin numbers specified in the PSEL.DFEGPIO[n] registers. Only pins that have the PSEL.DFEGPIO[n].CONNECTED field set to `Connected` will be controlled by RADIO. Pins that are disconnected will be controlled by GPIO.

During transmission in AoD TX mode or reception in AoA RX mode, RADIO automatically acquires the pins as needed. At times when RADIO does not use the pin, the pin is released to its default state and controlled by the GPIO configuration. Thus, the pin must be configured using the GPIO peripheral.

Pin acquired by RADIO	Direction	Value	Comment
Yes	Output	Specified in SWITCHPATTERN	Pin acquired by RADIO, and in use for DFE.
No	Specified by GPIO	Specified by GPIO	DFE not in progress. Pin has not been acquired by RADIO, but is available for DFE use.

Table 125: Pin configuration matrix for a connected and enabled pin [n]

Switch pattern configuration

The values of the GPIOs while switching during the CTE are configured by writing successively to the SWITCHPATTERN register. The first write to SWITCHPATTERN is the GPIO pattern applied from the call of TASKS_TXEN or TASKS_RXEN until the first antenna switch is triggered. The second write sets the pattern for the reference period and is applied at the start of the guard period. The following writes set the pattern for the remaining switch slots and are applied at the start of each switch slot. If writing beyond the total number of antenna slots, the pattern will wrap to SWITCHPATTERN[2] and start over again. During operation, when the end of the SWITCHPATTERN buffer is reached, RADIO cycles back to SWITCHPATTERN[2]. At the end of the AoA/AoD procedure, SWITCHPATTERN[0] is applied to DFECTRL1.TSWITCHSPACING after the previous antenna switch. The SWITCHPATTERN buffer can be erased/cleared using CLEARPATTERN.

A minimum number of three patterns must be written to the SWITCHPATTERN register.

If CTEINLINECONF.CTEINLINECTRLEN is not set, then the antenna switch spacing is determined by DFECTRL1.TSWITCHSPACING (see Switching and sampling spacing based on CTEType on page 433). DFECTRL2.TSWITCHOFFSET determines the position of the first switch compared to the configurable start of CTE (see DFECTRL1.DFEINEXTENSION).

7.26.12.7 IQ sampling

RADIO uses DMA to write IQ samples recorded during the CTE to RAM. Alternatively, the magnitude and phase of the samples can be recorded using the DFECTRL1.SAMPLETYPE field. The samples are written to the location in RAM specified by DFEPACKET.PTR. The maximum number of samples to transfer are specified by DFEPACKET.MAXCNT and the number of samples transferred are given in DFEPACKET.AMOUNT. The IQ samples are recorded with respect to the RX carrier frequency. The format of the samples is provided in the following table.

SAMPLETYPE	Field	Bits	Description
0: I_Q (default)	Q	31:16	12 bits signed, sign extended to 16 bits. Out of range samples are saturated at value -32768.
	I	15:0	
1: MagPhase	reserved	31:29	Always zero
	magnitude	28:16	13 bits unsigned. Equals $1.646756*\sqrt{I^2 + Q^2}$.
	phase	15:0	9 bits signed, sign extended to 16 bits. Equals 64*atan2(Q, I) in the range [-201,201].

Table 126: Format of samples

Oversampling is configured separately for the reference period and for the time after the reference period. During the reference period, the sample spacing is determined by DFECTRL1.TSAMPLESPACINGREF. DFECTRL2.TSAMPLEOFFSET allows fine tuning the position of the samples in steps of 16 MHz periods (62.5 ns)

For the time after the reference period, if CTEINLINECONF.CTEINLINECTRLEN is disabled, the sample spacing is set in DFECTRL1.TSAMPLESPACING. However, when CTEINLINECONF.CTEINLINECTRLEN is enabled, the sample spacing is determined by two different registers, depending on whether the device is in AoA or AoD RX-mode.

For AoD RX mode, the sample spacing after the reference period is determined by the CTEType in the packet, as listed in the following table.

CTEType	Sample spacing
AoD 1 µs slots	CTEINLINECONF.CTEINLINERXMODE1US
AoD 2 µs slots	CTEINLINECONF.CTEINLINERXMODE2US
Other	DFECTRL1.TSAMPLESPACING

Table 127: Sample spacing when CTEINLINECONF.CTEINLINECTRLEN is set and the device is in AoD RX mode

For AoA RX mode, the sample spacing after the reference period is determined by DFECTRL1.TSWITCHSPACING, as listed in the following table.

DFECTRL1.TSWITCHSPACING	Sample spacing
2 µs	CTEINLINECONF.CTEINLINERXMODE1US
4 µs	CTEINLINECONF.CTEINLINERXMODE2US
Other	DFECTRL1.TSAMPLESPACING

Table 128: Sample spacing when CTEINLINECONF.CTEINLINECTRLEN is set and the device is in AoA RX mode

For the reference and switching periods, DFECTRL1.TSAMPLESPACINGREF and DFECTRL1.TSAMPLESPACING can be used to achieve oversampling.

7.26.13 IEEE 802.15.4 operation

With the MODE=Ieee802154_250kbit, RADIO will comply with the IEEE 802.15.4-2006 standard implementing its 250 kbps, 2450 MHz, O-QPSK PHY.

The IEEE 802.15.4 standard differs from Nordic's proprietary and Bluetooth Low Energy modes. Notable differences include modulation scheme, channel structure, packet structure, security, and medium access control.

The main features of the IEEE 802.15.4 mode are:

- Ultra-low power 250 kbps, 2450 MHz, IEEE 802.15.4-2006 compliant link
- Clear channel assessment (CCA)
- Energy detection (ED) scan
- CRC generation

7.26.13.1 Packet structure

The IEEE 802.15.4 standard defines an on-the-air frame/packet that is different from what is used in Bluetooth Low Energy.

The following figure provides an overview of the physical frame structure and its timing.

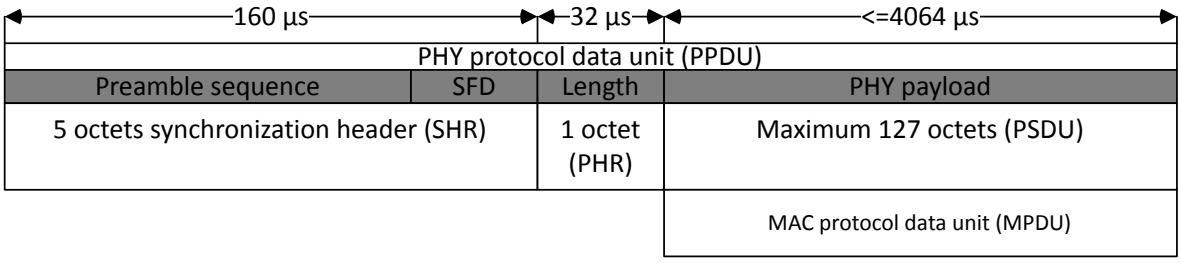

Figure 161: IEEE 802.15.4 frame format (PPDU)

The standard uses the term *octet* for an 8-bit storage unit within the PPDU. For timing, the value *symbol* is used, and it has a duration of 16 µs.

The total usable payload (PSDU) is 127 octets, but when CRC is in use, this is reduced to 125 octets of usable payload.

The preamble sequence consists of four octets that are all zero, and are used for synchronizing RADIO's receiver. Following the preamble is the single octet *start of frame delimiter (SFD)*, with a fixed value of 0xA7. An alternate SFD can be programmed through the SFD register, providing an initial level of frame filtering for those who choose non-standard compliance. It is a valuable feature when operating in a congested or private network. The preamble sequence and the SFD are generated by RADIO, and are not programmed by the user into the frame buffer.

Following the five octet *synchronization header (SHR)* is the single octet *phy header (PHR)*. The least significant seven bits of PHR denote the frame length of the following PSDU. The most significant bit is reserved and is set to 0 for frames that are standard compliant. RADIO reports all eight bits which can be used to carry additional information. The PHR is the first byte written to the frame data memory pointed to by PACKETPTR. Frames with zero length are discarded, and the FRAMESTART event is not generated in this case.

The next N octets carry the data of the PHY packet, where N equals the value of the PHR. For an implementation also using the IEEE 802.15.4 medium access control (MAC) layer, the PHY data is a MAC frame of N-2 octets, since two octets occupy a CRC field.

An IEEE 802.15.4 MAC layer frame consists of the following:

- A header:
  - The frame control field (FCF)
  - The sequence number
  - Addressing fields
- A payload
- The 16-bit frame control sequence (FCS)

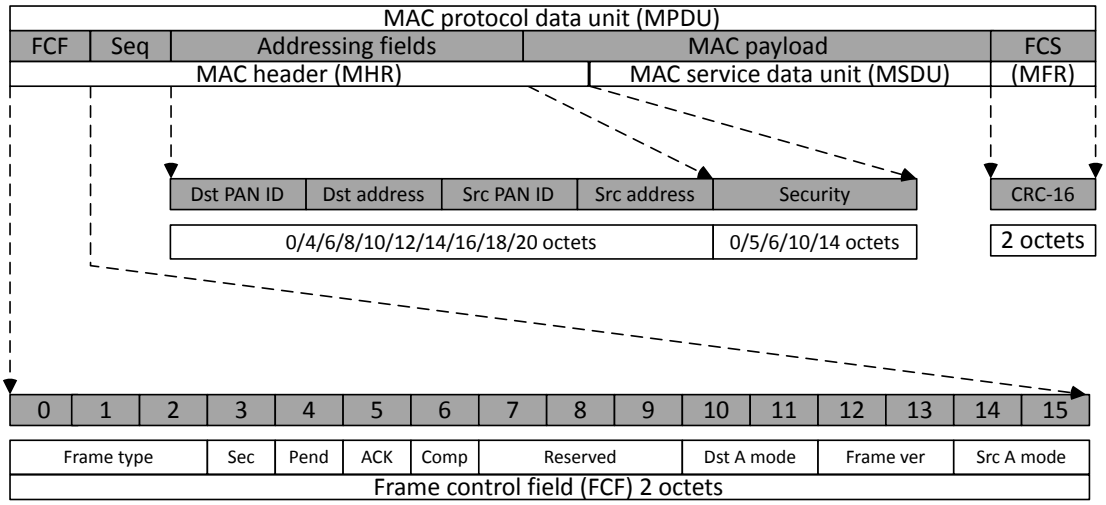

Figure 162: IEEE 802.15.4 frame format (MPDU)

The two FCF octets contain information about the frame type, addressing, and other control flags. This field is decoded when using the assisted operating modes offered by RADIO.

The sequence number is a single octet in size and is unique for a frame. It is used in the associated acknowledgement frame sent upon successful frame reception.

The addressing field can be zero (acknowledgement frame) or up to 20 octets in size. The field is used to direct packets to the correct recipient and denote its origin. IEEE 802.15.4 bases its addressing on networks being organized in PANs with 16-bit identifier and nodes having a 16-bit or 64-bit address. In the assisted receive mode, these parameters are analyzed for address matching and acknowledgement.

The MAC payload carries the data of the next higher layer, or in the case of a MAC command frame, information used by the MAC layer itself.

The two last octets contain the 16-bit ITU-T CRC. The FCS is calculated over the MAC header (MHR) and MAC payload (MSDU) parts of the frame. This field is calculated automatically when sending a frame, or indicated in the CRCSTATUS register when a frame is received. If configured, this feature is maintained autonomously by the CRC module.

7.26.13.2 Operating frequencies

The IEEE 802.15.4 standard defines 16 channels in the 2450 MHz frequency band. The channels are numbered from 11 to 26, and each channel is 5 MHz wide.

To choose the correct channel center frequency, the FREQUENCY register must be programmed according to the following table.

IEEE 802.15.4 channel	Center frequency (MHz)	FREQUENCY setting
Channel 11	2405	5
Channel 12	2410	10
Channel 13	2415	15
Channel 14	2420	20
Channel 15	2425	25
Channel 16	2430	30
Channel 17	2435	35
Channel 18	2440	40
Channel 19	2445	45
Channel 20	2450	50
Channel 21	2455	55
Channel 22	2460	60
Channel 23	2465	65
Channel 24	2470	70
Channel 25	2475	75
Channel 26	2480	80

Table 129: IEEE 802.15.4 center frequency definition

7.26.13.3 Energy detection (ED)

As required by the IEEE 802.15.4 standard, it must be possible to sample the received signal power within the bandwidth of a channel, for the purpose of determining presence of activity.

To prevent the channel signal from being decoded, the shortcut between the READY event and the START task should be disabled before putting RADIO in receive mode. The energy detection (ED) measurement time, where RSSI samples are averaged, is 8 symbol periods, corresponding to 128 µs. The standard further specifies the measurement to be a number between 0 and 255, where 0 shall indicate received power less than 10 dB above the selected receiver sensitivity. The power range of the ED values must be at least a 40 dB linear mapping with accuracy of ±6 dB. See section *6.9.7 Receiver ED* in the IEEE 802.15.4 standard for further details.

The following example shows how to perform a single energy detection measurement and convert to IEEE 802.15.4 scale.

IEEE 802.15.4 ED measurement example

```
#define ED_RSSISCALE 4 // From electrical specifications
uint8_t sample_ed(void)
{
    int val;
    NRF_RADIO->TASKS_EDSTART = 1; // Start
    while (NRF_RADIO->EVENTS_EDEND != 1) {
        // CPU can sleep here or do something else
        // Use of interrupts are encouraged
        }
    val = NRF_RADIO->EDSAMPLE * ED_RSSISCALE; // Read level
    return (uint8_t)(val>255 ? 255 : val);    // Convert to IEEE 802.15.4 scale
}
```

For scaling between hardware value and dBm, see equation Conversion between hardware value and dBm on page 441.

The mlme-scan.req primitive of the MAC layer uses the ED measurement to detect channels where there might be wireless activity. To assist this primitive, a tailored mode of operation is available where the ED measurement runs for a defined number of iterations keeping track of the maximum ED level. This is enganged by writing the EDCNT register to a value different from 0, where it will run the specified number of iterations and report the maximum energy measurement in the EDSAMPLE register. The scan is started with EDSTART task and its end indicated with the EDEND event. This significantly reduces the interrupt frequency and therefore power consumption. The following figure shows how the ED measurement will operate depending on the EDCNT register.

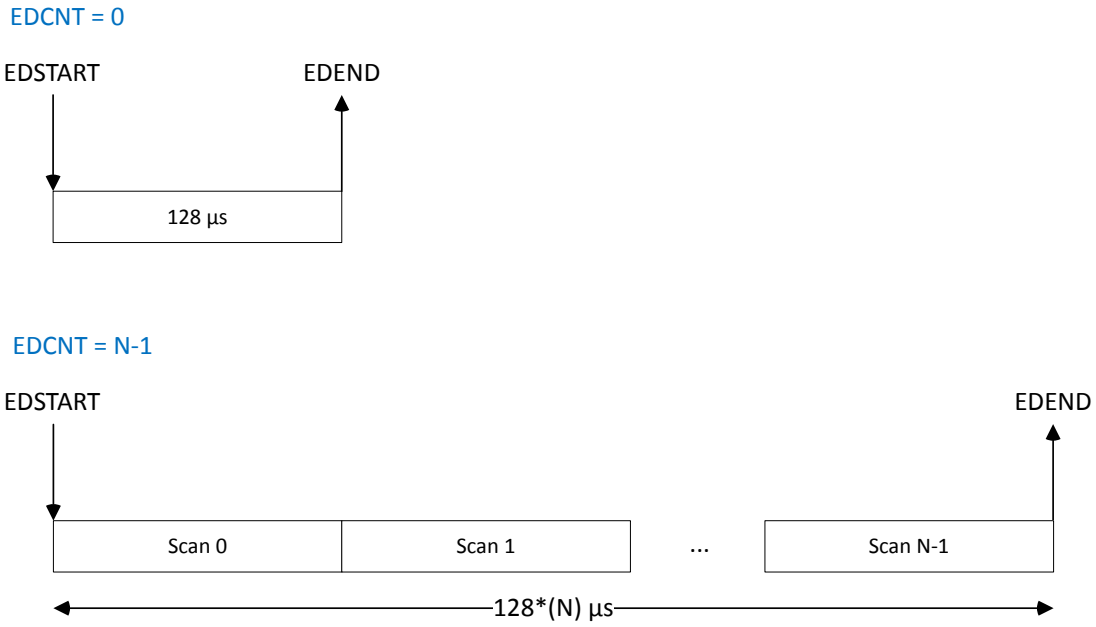

Figure 163: Energy detection measurement examples

The scan is stopped by writing the EDSTOP task. It is followed by the EDSTOPPED event when the module has terminated.

7.26.13.4 Clear channel assessment (CCA)

The IEEE 802.15.4 standard implements a listen-before-talk channel access method to avoid collisions when transmitting, known as *carrier sense multiple access with collision avoidance (CSMA-CA)*. The key part of this is measuring if the wireless medium is busy or not.

The following clear channel assesment modes are supported:

- *CCA Mode 1* (energy above threshold) - The medium is reported busy upon detecting any energy above the ED threshold.
- *CCA Mode 2* (carrier sense only) - The medium is reported busy upon detection of a signal compliant with the IEEE 802.15.4 standard with the same modulation and spreading characteristics.
- *CCA Mode 3* (carrier sense with energy above threshold) - The medium is reported busy using a logical combination (AND/OR) between the results from CCA Mode 1 and CCA Mode 2.

The clear channel assessment should survey a period equal to 8 symbols or 128 µs.

RADIO must be in RX mode and be able to receive correct packets when performing the CCA. The shortcut between READY and START must be disabled if baseband processing is not to be performed while the measurement is running.

CCA Mode 1

CCA Mode 1 is enabled by first configuring the field CCACTRL.CCAMODE=EdMode and writing the CCACTRL.CCAEDTHRES field to a chosen value. Once the CCASTART task is written, RADIO will perform an ED measurement for 8 symbols and compare the measured level with that found in the CCACTRL.CCAEDTHRES field. If the measured value is higher than or equal to this threshold, the CCABUSY event is generated. If the measured level is less than the threshold, the CCAIDLE event is generated.

CCA Mode 2

CCA Mode 2 is enabled by configuring CCACTRL.CCAMODE=CarrierMode. RADIO will sample to see if a valid SFD is found during the 8 symbols. If a valid SFD is detected, the CCABUSY event is generated and the device should not send any data. The CCABUSY event is also generated if the scan was performed during an ongoing frame reception. In the case where the measurement period completes with no SFD detection, the CCAIDLE event is generated. When CCACTRL.CCACORRCNT is not zero, the algorithm will look at the correlator output in addition to the SFD detection signal. If a SFD is reported during the scan period, it will terminate immidiately indicating busy medium. Similarly, if the number of peaks above CCACTRL.CCACORRTHRES crosses the CCACTRL.CCACORRCNT, the CCACTRL.CCABUSY event is generated. If less than CCACORRCOUNT crossings are found and no SFD is reported, the CCAIDLE event will be generated and the device can send data.

CCA Mode 3

CCA Mode 3 is enabled by configuring CCACTRL.CCAMODE=CarrierAndEdMode or CCACTRL.CCAMODE=CarrierOrEdMode, performing the required logical combination of the result from CCA Mode 1 and 2. The CCABUSY or CCAIDLE events are generated by ANDing or ORing the *energy above threshold* and *carrier detection* scans.

Shortcuts

An ongoing CCA can always be stopped by issuing the CCASTOP task. This will trigger the associated CCASTOPPED event.

For CCA mode automation, the following shortcuts are available:

- To automatically switch between RX mode (when performing the CCA) and to TX mode where the packet is sent, the shortcut between CCAIDLE and TXEN, in conjunction with the short between CCAIDLE and STOP must be used.
- To automatically disable RADIO whenever the CCA reports a busy medium, the shortcut between CCABUSY and DISABLE can be used.
- To immediately start a CCA after ramping up into RX mode, the shortcut between RXREADY and CCASTART can be used.

Conversion

The conversion from a CCAEDTHRES, LQI, or EDSAMPLE value to dBm can be done with the following equation, where $VAL_{HARDWARE}$ is either CCAEDTHRES, LQI, or EDSAMPLE. LQI and EDSAMPLE are hardware-reported values, while CCAEDTHRES is set by software. Constants ED_RSSISCALE and ED_RSSIOFFS are from electrical specifications.

$$P_{RF}[dBm] = ED\_RSSIOFFS + VAL_{HARDWARE}$$

Figure 164: Conversion between hardware value and dBm

The ED_RSSISCALE constant is used to calculate power in 802.15.4 units (0-255):

$$P_{RF}[802.15.4\ units] = MIN(\ ED\_RSSISCALE \times VAL_{HARDWARE},\ 255\ )$$

Figure 165: Conversion between hardware value and 802.15.4 units (0-255)

7.26.13.5 Cyclic redundancy check (CRC)

The IEEE 802.15.4 standard uses a 16-bit ITU-T cyclic redundancy check (CRC) calculated over the MAC header (MHR) and MAC service data unit (MSDU).

The standard defines the following generator polynomial:

$$G(x) = x^{16} + x^{12} + x^{5} + 1$$

In RX mode, RADIO will trigger the CRC module when the first octet after the frame length (PHR) is received. The CRC will then update on each consecutive octet received. When a complete frame is received the CRCSTATUS register will be updated accordingly and the CRCOK or CRCERROR events generated. When the CRC module is enabled it will not write the two last octets (CRC) to the frame Data RAM. When transmitting, the CRC will be computed on the fly, starting with the first octet after PHR, and inserted as the two last octets in the frame. The EasyDMA will fetch frame length minus 2 octets from RAM and insert the CRC octets at their correct positions in the frame.

The following code shows how to configure the CRC module for correct operation when in IEEE 802.15.4 mode. The CRCCNF is written to 16-bit CRC and the CRCPOLY is written to `0x11021`. The start value used by IEEE 802.15.4 is `0` and CRCINIT is configured to reflect this.

```
/* 16-bit CRC with ITU-T polynomial with 0 as start condition*/
NRF_RADIO->CRCCNF = ((RADIO_CRCCNF_SKIPADDR_Ieee802154 << RADIO_CRCCNF_SKIPADDR_Pos) |
                     (RADIO_CRCCNF_LEN_Two << RADIO_CRCCNF_LEN_Pos));
NRF_RADIO->CRCPOLY = 0x11021;
NRF_RADIO->CRCINIT = 0;
```

The ENDIANESS subregister must be set to little-endian since the FCS field is transmitted from left bit to right.

7.26.13.6 Transmit sequence

The transmission is started by first putting RADIO in RX mode and triggering the RXEN task.

An outline of the IEEE 802.15.4 transmission is illustrated in the following figure.

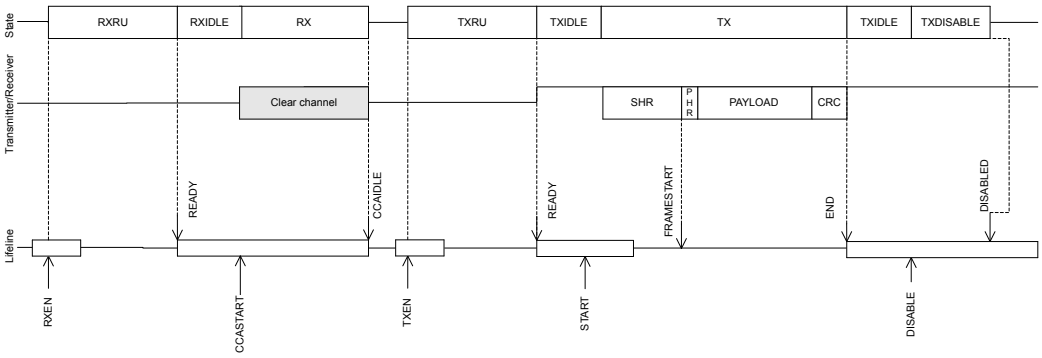

Figure 166: IEEE 802.15.4 transmit sequence

The receiver will ramp up and enter the RXIDLE state where the READY event is generated. Upon receiving the ready event, the CCA is started by triggering the CCASTART task. The chosen mode of assessment (CCACTRL.CCAMODE register) will be performed and signal the CCAIDLE or CCABUSY event 128 µs later. If the CCABUSY event is received, RADIO will have to retry the CCA after a specific back-off period. This is outlined in the *IEEE 802.15.4 standard, Figure 69 in section 7.5.1.4 The CSMA-CA algorithm*.

If the CCAIDLE event is generated, a write to the TXEN task register enters RADIO in TXRU state. The READY event will be generated when RADIO is in TXIDLE state and ready to transmit. With the PACKETPTR pointing to the length (PHR) field of the frame, the START task can be written. RADIO will send the four octet preamble sequence followed by the start of frame delimiter (SFD register). The first byte read from the Data RAM is the length field (PHR) followed by the transmission of the number of bytes indicated as the frame length. If the CRC module is configured it will run for PHR-2 octets. The last two octets will be substituted with the results from running the CRC. The necessary CRC parameters are sampled on the START task. The FCS field of the frame is little endian.

In addition to the already available shortcuts, one is provided between the READY event and the CCASTART task so that a CCA can automatically start when the receiver is ready. A second shortcut has been added between the CCAIDLE event and the TXEN task, so that upon detecting a clear channel RADIO can immediately enter TX mode.

7.26.13.7 Receive sequence

The reception is started by first putting RADIO in receive mode. After writing to the RXEN task, RADIO will start ramping up and enter the RXRU state.

When the READY event is generated, RADIO enters the RXIDLE mode. For the baseband processing to be enabled, the START task must be written. An outline of the IEEE 802.15.4 reception can be found in the following figure.

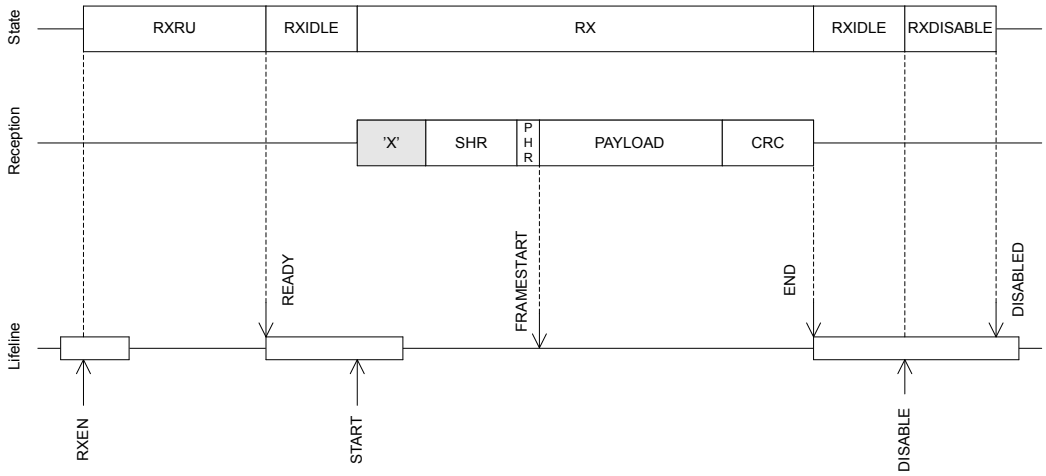

Figure 167: IEEE 802.15.4 receive sequence

When a valid SHR is received, RADIO will start storing future octets (starting with PHR) to the data memory pointed to by PACKETPTR. After the SFD octet is received, the FRAMESTART event is generated. If the CRC module is enabled it will start updating with the second byte received (first byte in payload) and run for the full frame length. The two last bytes in the frame are not written to RAM when CRC is configured. However, if the result of the CRC after running the full frame is zero, the CRCOK event will be generated. The END event is generated when the last octet has been received and is available in data memory.

When a packet is received, a link quality indicator (LQI) is also generated and appended immediately after the last received octet. When using an IEEE 802.15.4 compliant frame, this will be just after the MSDU since the FCS is not reported. In the case of a non-compliant frame, it will be appended after the full frame. The LQI reported by the hardware must be converted to the IEEE 802.15.4 range by an 8-bit saturating multiplication of 4, as shown in IEEE 802.15.4 ED measurement example on page 439. The LQI is only valid for frames equal to or longer than three octets. When receiving a frame, the RSSI (reported as negative dB) will be measured at three points during the reception. These three values will be sorted and the middle one selected (median 3) to be remapped within the LQI range. The following figure illustrates the LQI measurement and how the data is arranged in data memory.

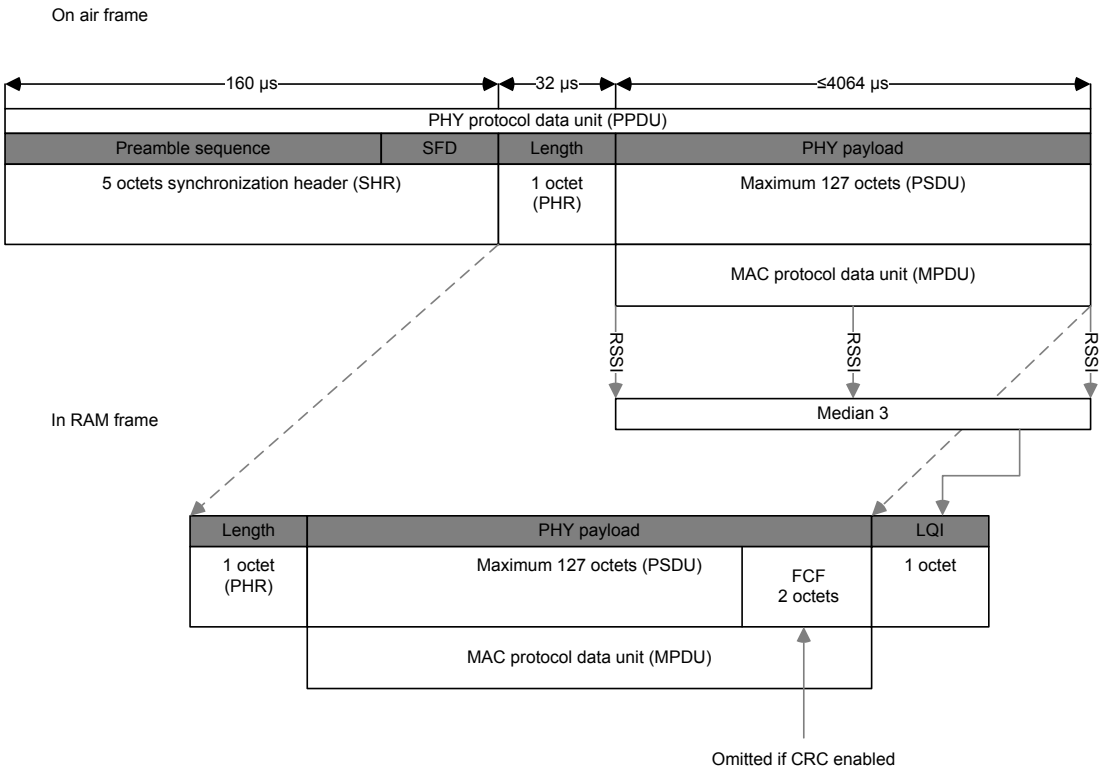

Figure 168: IEEE 802.15.4 frame in data memory

A shortcut has been added between the FRAMESTART event and the BCSTART task. This can be used to trigger a BCMATCH event after N bits, such as when inspecting the MAC addressing fields.

7.26.13.8 Interframe spacing (IFS)

The IEEE 802.15.4 standard defines a specific time that is alotted for the MAC sublayer to process received data. The interframe spacing (IFS) is used to prevent two frames from being transmitted too close together. If the transmission is requesting an acknowledgement, the space before the second frame shall be at least one IFS period.

IFS is determined to be one of the following:

- IFS equals macMinSIFSPeriod (12 symbols) if the MPDU is less than or equal to aMaxSIFSFrameSize (18 octets) octets
- IFS equals macMinLIFSPeriod (40 symbols) if the MPDU is larger than aMaxSIFSFrameSize

Using the efficient assisted modes in RADIO, the TIFS will be programmed with the correct value based on the frame being transmitted. If the assisted modes are not in use, the TIFS register must be updated manually. The following figure provides details on what IFS period is valid in both acknowledged and unacknowledged transmissions.

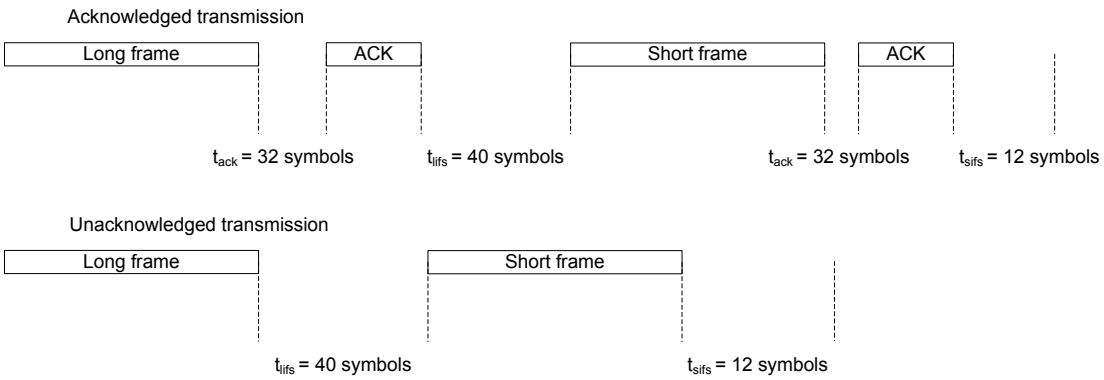

Figure 169: Interframe spacing examples

7.26.14 EasyDMA

RADIO uses EasyDMA to read and write packets to RAM without CPU involvement.

As illustrated in RADIO block diagram on page 422, RADIO's EasyDMA utilizes the same PACKETPTR for receiving and transmitting packets. This pointer should be reconfigured by the CPU each time before RADIO is started by the START task. The PACKETPTR register is double-buffered, meaning that it can be updated and prepared for the next transmission.

The END event indicates that the last bit has been processed by RADIO. The DISABLED event is issued to acknowledge that a DISABLE task is done.

The structure of a packet is described in detail in Packet configuration on page 423. The data that is stored in Data RAM and transported by EasyDMA consists of the following fields:

- S0
- LENGTH
- S1
- PAYLOAD

In addition, a static add-on is sent immediately after the payload.

The size of each of the above fields in the frame is configurable (see Packet configuration on page 423), and the space occupied in RAM depends on these settings. The size of the field can be zero, as long as the resulting frame complies with the chosen RF protocol.

All fields are extended in size to align with a byte boundary in RAM. For instance, a 3-bit long field on air will occupy 1 byte in RAM while a 9-bit long field will be extended to 2 bytes.

The packet's elements can be configured as follows:

- CI, TERM1, and TERM2 fields are only present in Bluetooth Low Energy Long Range mode
- S0 is configured through the PCNF0.S0LEN field
- LENGTH is configured through the PCNF0.LFLEN field
- S1 is configured through the PCNF0.S1LEN field
- Payload size is configured through the value in RAM corresponding to the LENGTH field
- Static add-on size is configured through the PCNF1.STATLEN field

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

Note: The PCNF1.MAXLEN field includes the payload and the add-on, but excludes the size occupied by the S0, LENGTH, and S1 fields. This has to be taken into account when allocating RAM.

If the payload and add-on length is specified larger than PCNF1.MAXLEN, RADIO will still transmit or receive in the same way as before, except the payload is now truncated to PCNF1.MAXLEN. The packet's LENGTH field will not be altered when the payload is truncated. RADIO will calculate CRC as if the packet length is equal to PCNF1.MAXLEN.

Note: If PACKETPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 18 for more information about the different memory regions.

The END event indicates that the last bit has been processed by RADIO. The DISABLED event is issued to acknowledge that an DISABLE task is done.

7.26.15 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x41008000	NETWORK	RADIO	RADIO	NS	NA	2.4 GHz radio	

Table 130: Instances

Register	Offset	Security	Description
TASKS_TXEN	0x000		Enable RADIO in TX mode
TASKS_RXEN	0x004		Enable RADIO in RX mode
TASKS_START	0x008		Start RADIO
TASKS_STOP	0x00C		Stop RADIO
TASKS_DISABLE	0x010		Disable RADIO
TASKS_RSSISTART	0x014		Start the RSSI and take one single sample of the receive signal strength
TASKS_RSSISTOP	0x018		Stop the RSSI measurement
TASKS_BCSTART	0x01C		Start the bit counter
TASKS_BCSTOP	0x020		Stop the bit counter
TASKS_EDSTART	0x024		Start the energy detect measurement used in IEEE 802.15.4 mode
TASKS_EDSTOP	0x028		Stop the energy detect measurement
TASKS_CCASTART	0x02C		Start the clear channel assessment used in IEEE 802.15.4 mode
TASKS_CCASTOP	0x030		Stop the clear channel assessment
SUBSCRIBE_TXEN	0x080		Subscribe configuration for task TXEN
SUBSCRIBE_RXEN	0x084		Subscribe configuration for task RXEN
SUBSCRIBE_START	0x088		Subscribe configuration for task START
SUBSCRIBE_STOP	0x08C		Subscribe configuration for task STOP
SUBSCRIBE_DISABLE	0x090		Subscribe configuration for task DISABLE
SUBSCRIBE_RSSISTART	0x094		Subscribe configuration for task RSSISTART
SUBSCRIBE_RSSISTOP	0x098		Subscribe configuration for task RSSISTOP
SUBSCRIBE_BCSTART	0x09C		Subscribe configuration for task BCSTART
SUBSCRIBE_BCSTOP	0x0A0		Subscribe configuration for task BCSTOP
SUBSCRIBE_EDSTART	0x0A4		Subscribe configuration for task EDSTART
SUBSCRIBE_EDSTOP	0x0A8		Subscribe configuration for task EDSTOP
SUBSCRIBE_CCASTART	0x0AC		Subscribe configuration for task CCASTART
SUBSCRIBE_CCASTOP	0x0B0		Subscribe configuration for task CCASTOP
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

Register	Offset	Security	Description
EVENTS_DISABLED	0x110		RADIO has been disabled
EVENTS_DEVMATCH	0x114		A device address match occurred on the last received packet
EVENTS_DEVMISS	0x118		No device address match occurred on the last received packet
EVENTS_RSSIEND	0x11C		Sampling of receive signal strength complete
EVENTS_BCMATCH	0x128		Bit counter reached bit count value
EVENTS_CRCOK	0x130		Packet received with CRC ok
EVENTS_CRCERROR	0x134		Packet received with CRC error
EVENTS_FRAMESTART	0x138		IEEE 802.15.4 length field received
EVENTS_EDEND	0x13C		Sampling of energy detection complete. A new ED sample is ready for readout from the RADIO.EDSAMPLE register.
EVENTS_EDSTOPPED	0x140		The sampling of energy detection has stopped
EVENTS_CCAIDLE	0x144		Wireless medium in idle - clear to send
EVENTS_CCABUSY	0x148		Wireless medium busy - do not send
EVENTS_CCASTOPPED	0x14C		The CCA has stopped
EVENTS_RATEBOOST	0x150		Ble_LR CI field received, receive mode is changed from Ble_LR125Kbit to Ble_LR500Kbit.
EVENTS_TXREADY	0x154		RADIO has ramped up and is ready to be started TX path
EVENTS_RXREADY	0x158		RADIO has ramped up and is ready to be started RX path
EVENTS_MHRMATCH	0x15C		MAC header match found
EVENTS_SYNC	0x168		Preamble indicator
EVENTS_PHYEND	0x16C		Generated when last bit is sent on air, or received from air
EVENTS_CTEPRESENT	0x170		CTE is present (early warning right after receiving CTEInfo byte)
PUBLISH_READY	0x180		Publish configuration for event READY
PUBLISH_ADDRESS	0x184		Publish configuration for event ADDRESS
PUBLISH_PAYLOAD	0x188		Publish configuration for event PAYLOAD
PUBLISH_END	0x18C		Publish configuration for event END
PUBLISH_DISABLED	0x190		Publish configuration for event DISABLED
PUBLISH_DEVMATCH	0x194		Publish configuration for event DEVMATCH
PUBLISH_DEVMISS	0x198		Publish configuration for event DEVMISS
PUBLISH_RSSIEND	0x19C		Publish configuration for event RSSIEND
PUBLISH_BCMATCH	0x1A8		Publish configuration for event BCMATCH
PUBLISH_CRCOK	0x1B0		Publish configuration for event CRCOK
PUBLISH_CRCERROR	0x1B4		Publish configuration for event CRCERROR
PUBLISH_FRAMESTART	0x1B8		Publish configuration for event FRAMESTART
PUBLISH_EDEND	0x1BC		Publish configuration for event EDEND
PUBLISH_EDSTOPPED	0x1C0		Publish configuration for event EDSTOPPED
PUBLISH_CCAIDLE	0x1C4		Publish configuration for event CCAIDLE
PUBLISH_CCABUSY	0x1C8		Publish configuration for event CCABUSY
PUBLISH_CCASTOPPED	0x1CC		Publish configuration for event CCASTOPPED
PUBLISH_RATEBOOST	0x1D0		Publish configuration for event RATEBOOST
PUBLISH_TXREADY	0x1D4		Publish configuration for event TXREADY
PUBLISH_RXREADY	0x1D8		Publish configuration for event RXREADY
PUBLISH_MHRMATCH	0x1DC		Publish configuration for event MHRMATCH
PUBLISH_SYNC	0x1E8		Publish configuration for event SYNC
PUBLISH_PHYEND	0x1EC		Publish configuration for event PHYEND
PUBLISH_CTEPRESENT	0x1F0		Publish configuration for event CTEPRESENT
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
CRCSTATUS	0x400		CRC status
RXMATCH	0x408		Received address
RXCRC	0x40C		CRC field of previously received packet
DAI	0x410		Device address match index

Register	Offset	Security	Description
PDUSTAT	0x414		Payload status
CTESTATUS	0x44C		CTEInfo parsed from received packet
DFESTATUS	0x458		DFE status information
PACKETPTR	0x504		Packet pointer
FREQUENCY	0x508		Frequency
TXPOWER	0x50C		Output power
MODE	0x510		Data rate and modulation
PCNF0	0x514		Packet configuration register 0
PCNF1	0x518		Packet configuration register 1
BASE0	0x51C		Base address 0
BASE1	0x520		Base address 1
PREFIX0	0x524		Prefixes bytes for logical addresses 0-3
PREFIX1	0x528		Prefixes bytes for logical addresses 4-7
TXADDRESS	0x52C		Transmit address select
RXADDRESSES	0x530		Receive address select
CRCCNF	0x534		CRC configuration
CRCPOLY	0x538		CRC polynomial
CRCINIT	0x53C		CRC initial value
TIFS	0x544		Interframe spacing in µs
RSSISAMPLE	0x548		RSSI sample
STATE	0x550		Current radio state
DATAWHITEIV	0x554		Data whitening initial value
BCC	0x560		Bit counter compare
DAB[n]	0x600		Device address base segment n
DAP[n]	0x620		Device address prefix n
DACNF	0x640		Device address match configuration
MHRMATCHCONF	0x644		Search pattern configuration
MHRMATCHMAS	0x648		Pattern mask
MODECNF0	0x650		Radio mode configuration register 0
SFD	0x660		IEEE 802.15.4 start of frame delimiter
EDCNT	0x664		IEEE 802.15.4 energy detect loop count
EDSAMPLE	0x668		IEEE 802.15.4 energy detect level
CCACTRL	0x66C		IEEE 802.15.4 clear channel assessment control
DFEMODE	0x900		Whether to use Angle-of-Arrival (AOA) or Angle-of-Departure (AOD)
CTEINLINECONF	0x904		Configuration for CTE inline mode
DFECTRL1	0x910		Various configuration for Direction finding
DFECTRL2	0x914		Start offset for Direction finding
SWITCHPATTERN	0x928		GPIO patterns to be used for each antenna
CLEARPATTERN	0x92C		Clear the GPIO pattern array for antenna control
PSEL.DFEGPIO[n]	0x930		Pin select for DFE pin n
DFEPACKET.PTR	0x950		Data pointer
DFEPACKET.MAXCNT	0x954		Maximum number of buffer words to transfer
DFEPACKET.AMOUNT	0x958		Number of samples transferred in the last transaction
POWER	0xFFC		Peripheral power control

Table 131: Register overview

7.26.15.1 TASKS_TXEN

Address offset: 0x000

Enable RADIO in TX mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_TXEN			Enable RADIO in TX mode
			Trigger	1	Trigger task

7.26.15.2 TASKS_RXEN

Address offset: 0x004

Enable RADIO in RX mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_RXEN			Enable RADIO in RX mode
			Trigger	1	Trigger task

7.26.15.3 TASKS_START

Address offset: 0x008

Start RADIO

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start RADIO
			Trigger	1	Trigger task

7.26.15.4 TASKS_STOP

Address offset: 0x00C

Stop RADIO

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop RADIO
			Trigger	1	Trigger task

7.26.15.5 TASKS_DISABLE

Address offset: 0x010

Disable RADIO

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_DISABLE			Disable RADIO
			Trigger	1	Trigger task

7.26.15.6 TASKS_RSSISTART

Address offset: 0x014

Start the RSSI and take one single sample of the receive signal strength

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RSSISTART			Start the RSSI and take one single sample of the receive signal strength
			Trigger	1	Trigger task

7.26.15.7 TASKS_RSSISTOP

Address offset: 0x018

Stop the RSSI measurement

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RSSISTOP			Stop the RSSI measurement
			Trigger	1	Trigger task

7.26.15.8 TASKS_BCSTART

Address offset: 0x01C

Start the bit counter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_BCSTART			Start the bit counter
			Trigger	1	Trigger task

7.26.15.9 TASKS_BCSTOP

Address offset: 0x020

Stop the bit counter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_BCSTOP			Stop the bit counter
			Trigger	1	Trigger task

7.26.15.10 TASKS_EDSTART

Address offset: 0x024

Start the energy detect measurement used in IEEE 802.15.4 mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_EDSTART			Start the energy detect measurement used in IEEE 802.15.4 mode
			Trigger	1	Trigger task

7.26.15.11 TASKS_EDSTOP

Address offset: 0x028

Stop the energy detect measurement

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_EDSTOP			Stop the energy detect measurement
			Trigger	1	Trigger task

7.26.15.12 TASKS_CCASTART

Address offset: 0x02C

Start the clear channel assessment used in IEEE 802.15.4 mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_CCASTART			Start the clear channel assessment used in IEEE 802.15.4 mode
			Trigger	1	Trigger task

7.26.15.13 TASKS_CCASTOP

Address offset: 0x030

Stop the clear channel assessment

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_CCASTOP			Stop the clear channel assessment
			Trigger	1	Trigger task

7.26.15.14 SUBSCRIBE_TXEN

Address offset: 0x080

Subscribe configuration for task TXEN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task TXEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.15 SUBSCRIBE_RXEN

Address offset: 0x084

Subscribe configuration for task RXEN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RXEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.16 SUBSCRIBE_START

Address offset: 0x088

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.17 SUBSCRIBE_STOP

Address offset: 0x08C

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.18 SUBSCRIBE_DISABLE

Address offset: 0x090

Subscribe configuration for task 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task DISABLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.19 SUBSCRIBE_RSSISTART

Address offset: 0x094

Subscribe configuration for task RSSISTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RSSISTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.20 SUBSCRIBE_RSSISTOP

Address offset: 0x098

Subscribe configuration for task RSSISTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RSSISTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.21 SUBSCRIBE_BCSTART

Address offset: 0x09C

Subscribe configuration for task BCSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task BCSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.22 SUBSCRIBE_BCSTOP

Address offset: 0x0A0

Subscribe configuration for task BCSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task BCSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.23 SUBSCRIBE_EDSTART

Address offset: 0x0A4

Subscribe configuration for task EDSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task EDSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.24 SUBSCRIBE_EDSTOP

Address offset: 0x0A8

Subscribe configuration for task EDSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task EDSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.25 SUBSCRIBE_CCASTART

Address offset: 0x0AC

Subscribe configuration for task CCASTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CCASTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.26 SUBSCRIBE_CCASTOP

Address offset: 0x0B0

Subscribe configuration for task CCASTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CCASTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.26.15.27 EVENTS_READY

Address offset: 0x100

RADIO has ramped up and is ready to be started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_READY			RADIO has ramped up and is ready to be started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.28 EVENTS_ADDRESS

Address offset: 0x104

Address sent or received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ADDRESS			Address sent or received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.29 EVENTS_PAYLOAD

Address offset: 0x108

Packet payload sent or received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_PAYLOAD			Packet payload sent or received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.30 EVENTS_END

Address offset: 0x10C

Packet sent or received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			Packet sent or received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.31 EVENTS_DISABLED

Address offset: 0x110

RADIO has been 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_DISABLED			RADIO has been disabled
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.32 EVENTS_DEVMATCH

Address offset: 0x114

A device address match occurred on the last received packet

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_DEVMATCH			A device address match occurred on the last received packet
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.33 EVENTS_DEVMISS

Address offset: 0x118

No device address match occurred on the last received packet

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_DEVMISS			No device address match occurred on the last received packet
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.34 EVENTS_RSSIEND

Address offset: 0x11C

Sampling of receive signal strength complete

A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RSSIEND			Sampling of receive signal strength complete A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.35 EVENTS_BCMATCH

Address offset: 0x128

Bit counter reached bit count value

Bit counter value is specified in the RADIO.BCC register

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_BCMATCH			Bit counter reached bit count value
					Bit counter value is specified in the RADIO.BCC register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.36 EVENTS_CRCOK

Address offset: 0x130

Packet received with CRC ok

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CRCOK			Packet received with CRC ok
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.37 EVENTS_CRCERROR

Address offset: 0x134

Packet received with CRC error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_CRCERROR			Packet received with CRC error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.38 EVENTS_FRAMESTART

Address offset: 0x138

IEEE 802.15.4 length field received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_FRAMESTART			IEEE 802.15.4 length field received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.39 EVENTS_EDEND

Address offset: 0x13C

Sampling of energy detection complete. A new ED sample is ready for readout from the RADIO.EDSAMPLE register.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_EDEND			Sampling of energy detection complete. A new ED sample is ready for readout from the RADIO.EDSAMPLE register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.40 EVENTS_EDSTOPPED

Address offset: 0x140

The sampling of energy detection has stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_EDSTOPPED			The sampling of energy detection has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.41 EVENTS_CCAIDLE

Address offset: 0x144

Wireless medium in idle - clear to send

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CCAIDLE			Wireless medium in idle - clear to send
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.42 EVENTS_CCABUSY

Address offset: 0x148

Wireless medium busy - do not send

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CCABUSY			Wireless medium busy - do not send
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.43 EVENTS_CCASTOPPED

Address offset: 0x14C

The CCA has stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_CCASTOPPED			The CCA has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.44 EVENTS_RATEBOOST

Address offset: 0x150

Ble_LR CI field received, receive mode is changed from Ble_LR125Kbit to Ble_LR500Kbit.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RATEBOOST			Ble_LR CI field received, receive mode is changed from Ble_LR125Kbit to Ble_LR500Kbit.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.45 EVENTS_TXREADY

Address offset: 0x154

RADIO has ramped up and is ready to be started TX path

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXREADY			RADIO has ramped up and is ready to be started TX path
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.46 EVENTS_RXREADY

Address offset: 0x158

RADIO has ramped up and is ready to be started RX path

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXREADY			RADIO has ramped up and is ready to be started RX path
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.47 EVENTS_MHRMATCH

Address offset: 0x15C

MAC header match found

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_MHRMATCH			MAC header match found
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.48 EVENTS_SYNC

Address offset: 0x168

Preamble indicator

A possible preamble has been received in Ble_LR125Kbit, Ble_LR500Kbit, or Ieee802154_250Kbit modes during an RX transaction. False triggering of the event is possible.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SYNC			Preamble indicator A possible preamble has been received in Ble_LR125Kbit, Ble_LR500Kbit, or Ieee802154_250Kbit modes during an RX transaction. False triggering of the event is possible.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.49 EVENTS_PHYEND

Address offset: 0x16C

Generated when last bit is sent on air, or received from air

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_PHYEND			Generated when last bit is sent on air, or received from air
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.50 EVENTS_CTEPRESENT

Address offset: 0x170

CTE is present (early warning right after receiving CTEInfo byte)

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CTEPRESENT			CTE is present (early warning right after receiving CTEInfo byte)
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.26.15.51 PUBLISH_READY

Address offset: 0x180

Publish configuration for event READY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.52 PUBLISH_ADDRESS

Address offset: 0x184

Publish configuration for event 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				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ADDRESS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.53 PUBLISH_PAYLOAD

Address offset: 0x188

Publish configuration for event PAYLOAD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event PAYLOAD will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.54 PUBLISH_END

Address offset: 0x18C

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.55 PUBLISH_DISABLED

Address offset: 0x190

Publish configuration for event 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DISABLED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.56 PUBLISH_DEVMATCH

Address offset: 0x194

Publish configuration for event DEVMATCH

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DEVMATCH will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.57 PUBLISH_DEVMISS

Address offset: 0x198

Publish configuration for event DEVMISS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DEVMISS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.58 PUBLISH_RSSIEND

Address offset: 0x19C

Publish configuration for event RSSIEND

A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RSSIEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.59 PUBLISH_BCMATCH

Address offset: 0x1A8

Publish configuration for event BCMATCH

Bit counter value is specified in the RADIO.BCC 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 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event BCMATCH will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.60 PUBLISH_CRCOK

Address offset: 0x1B0

Publish configuration for event CRCOK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CRCOK will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.61 PUBLISH_CRCERROR

Address offset: 0x1B4

Publish configuration for event CRCERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CRCERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.62 PUBLISH_FRAMESTART

Address offset: 0x1B8

Publish configuration for event FRAMESTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event FRAMESTART will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.63 PUBLISH_EDEND

Address offset: 0x1BC

Publish configuration for event EDEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event EDEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.64 PUBLISH_EDSTOPPED

Address offset: 0x1C0

Publish configuration for event EDSTOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event EDSTOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.65 PUBLISH_CCAIDLE

Address offset: 0x1C4

Publish configuration for event CCAIDLE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CCAIDLE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.66 PUBLISH_CCABUSY

Address offset: 0x1C8

Publish configuration for event CCABUSY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CCABUSY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.67 PUBLISH_CCASTOPPED

Address offset: 0x1CC

Publish configuration for event CCASTOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CCASTOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.68 PUBLISH_RATEBOOST

Address offset: 0x1D0

Publish configuration for event RATEBOOST

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RATEBOOST will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.69 PUBLISH_TXREADY

Address offset: 0x1D4

Publish configuration for event TXREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXREADY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.70 PUBLISH_RXREADY

Address offset: 0x1D8

Publish configuration for event RXREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXREADY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.71 PUBLISH_MHRMATCH

Address offset: 0x1DC

Publish configuration for event MHRMATCH

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event MHRMATCH will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.72 PUBLISH_SYNC

Address offset: 0x1E8

Publish configuration for event SYNC

A possible preamble has been received in Ble_LR125Kbit, Ble_LR500Kbit, or Ieee802154_250Kbit modes during an RX transaction. False triggering of the event is possible.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SYNC will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.73 PUBLISH_PHYEND

Address offset: 0x1EC

Publish configuration for event PHYEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event PHYEND will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.74 PUBLISH_CTEPRESENT

Address offset: 0x1F0

Publish configuration for event CTEPRESENT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CTEPRESENT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.26.15.75 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				U T S R Q P O N M L K H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	READY_START			Shortcut between event READY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	END_DISABLE			Shortcut between event END and task DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	DISABLED_TXEN			Shortcut between event DISABLED and task TXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	DISABLED_RXEN			Shortcut between event DISABLED and task RXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	ADDRESS_RSSISTART			Shortcut between event ADDRESS and task RSSISTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	END_START			Shortcut between event END and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G	RW	ADDRESS_BCSTART			Shortcut between event ADDRESS and task BCSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

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

ID	R/W	Field	Value ID	Value	Description
H	RW	DISABLED_RSSISTOP			Shortcut between event DISABLED and task RSSISTOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
K	RW	RXREADY_CCASTART			Shortcut between event RXREADY and task CCASTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
L	RW	CCAIDLE_TXEN			Shortcut between event CCAIDLE and task TXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
M	RW	CCABUSY_DISABLE			Shortcut between event CCABUSY and task DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
N	RW	FRAMESTART_BCSTART			Shortcut between event FRAMESTART and task BCSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
O	RW	READY_EDSTART			Shortcut between event READY and task EDSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
P	RW	EDEND_DISABLE			Shortcut between event EDEND and task DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
Q	RW	CCAIDLE_STOP			Shortcut between event CCAIDLE and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
R	RW	TXREADY_START			Shortcut between event TXREADY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
S	RW	RXREADY_START			Shortcut between event RXREADY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
T	RW	PHYEND_DISABLE			Shortcut between event PHYEND and task DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
U	RW	PHYEND_START			Shortcut between event PHYEND and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.26.15.76 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 Z Y V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to enable interrupt for event READY
			Set	1	Enable
			Disabled	0	Read: Disabled

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

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
B	RW	ADDRESS			Write '1' to enable interrupt for event ADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	PAYLOAD			Write '1' to enable interrupt for event PAYLOAD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	DISABLED			Write '1' to enable interrupt for event DISABLED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	DEVMATCH			Write '1' to enable interrupt for event DEVMATCH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	DEVMISS			Write '1' to enable interrupt for event DEVMISS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RSSIEND			Write '1' to enable interrupt for event RSSIEND A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	BCMATCH			Write '1' to enable interrupt for event BCMATCH Bit counter value is specified in the RADIO.BCC register
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CRCOK			Write '1' to enable interrupt for event CRCOK
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CRCERROR			Write '1' to enable interrupt for event CRCERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	FRAMESTART			Write '1' to enable interrupt for event FRAMESTART
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	EDEND			Write '1' to enable interrupt for event EDEND
			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							a	Z	Y			V	U	T	S	R	Q	P	O	N	M	L	K		I			H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
O	RW	EDSTOPPED										Write '1' to enable interrupt for event EDSTOPPED																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
P	RW	CCAIDLE										Write '1' to enable interrupt for event CCAIDLE																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
Q	RW	CCABUSY										Write '1' to enable interrupt for event CCABUSY																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
R	RW	CCASTOPPED										Write '1' to enable interrupt for event CCASTOPPED																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
S	RW	RATEBOOST										Write '1' to enable interrupt for event RATEBOOST																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
T	RW	TXREADY										Write '1' to enable interrupt for event TXREADY																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
U	RW	RXREADY										Write '1' to enable interrupt for event RXREADY																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
V	RW	MHRMATCH										Write '1' to enable interrupt for event MHRMATCH																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
Y	RW	SYNC										Write '1' to enable interrupt for event SYNC A possible preamble has been received in Ble_LR125Kbit, Ble_LR500Kbit, or Ieee802154_250Kbit modes during an RX transaction. False triggering of the event is possible.																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
Z	RW	PHYEND										Write '1' to enable interrupt for event PHYEND																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
a	RW	CTEPRESENT										Write '1' to enable interrupt for event CTEPRESENT																							
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							

7.26.15.77 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	Z	Y			V	U	T	S	R	Q	P	O	N	M	L	K		I			H	G	F	E	D	C	B	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ADDRESS			Write '1' to disable interrupt for event ADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	PAYLOAD			Write '1' to disable interrupt for event PAYLOAD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	DISABLED			Write '1' to disable interrupt for event DISABLED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	DEVMATCH			Write '1' to disable interrupt for event DEVMATCH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	DEVMISS			Write '1' to disable interrupt for event DEVMISS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RSSIEND			Write '1' to disable interrupt for event RSSIEND A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	BCMATCH			Write '1' to disable interrupt for event BCMATCH Bit counter value is specified in the RADIO.BCC register
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CRCOK			Write '1' to disable interrupt for event CRCOK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CRCERROR			Write '1' to disable interrupt for event CRCERROR

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID							a	Z	Y			V	U	T	S	R	Q	P	O	N	M	L	K		I			H	G	F	E	D	C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
M	RW	FRAMESTART										Write '1' to disable interrupt for event FRAMESTART																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
N	RW	EDEND										Write '1' to disable interrupt for event EDEND																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
O	RW	EDSTOPPED										Write '1' to disable interrupt for event EDSTOPPED																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
P	RW	CCAIDLE										Write '1' to disable interrupt for event CCAIDLE																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
Q	RW	CCABUSY										Write '1' to disable interrupt for event CCABUSY																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
R	RW	CCASTOPPED										Write '1' to disable interrupt for event CCASTOPPED																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
S	RW	RATEBOOST										Write '1' to disable interrupt for event RATEBOOST																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
T	RW	TXREADY										Write '1' to disable interrupt for event TXREADY																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
U	RW	RXREADY										Write '1' to disable interrupt for event RXREADY																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
V	RW	MHRMATCH										Write '1' to disable interrupt for event MHRMATCH																							
			Clear	1								Disable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							
Y	RW	SYNC										Write '1' to disable interrupt for event SYNC A possible preamble has been received in Ble_LR125Kbit, Ble_LR500Kbit, or Ieee802154_250Kbit modes during an RX transaction. False triggering of the event is possible.																							
			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				a Z Y V U T S R Q P O N M L K I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
Z	RW	PHYEND			Write '1' to disable interrupt for event PHYEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
a	RW	CTEPRESENT			Write '1' to disable interrupt for event CTEPRESENT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.26.15.78 CRCSTATUS

Address offset: 0x400

CRC status

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

ID	R/W	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

7.26.15.79 RXMATCH

Address offset: 0x408

Received address

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

ID	R/W	Field	Value ID	Value	Description
A	R	RXMATCH			Received address
					Logical address of which previous packet was received

7.26.15.80 RXCRC

Address offset: 0x40C

CRC field of previously received packet

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

ID	R/W	Field	Value ID	Value	Description
A	R	RXCRC			CRC field of previously received packet
					CRC field of previously received packet

7.26.15.81 DAI

Address offset: 0x410

Device address match index

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

ID	R/W	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

7.26.15.82 PDUSTAT

Address offset: 0x414

Payload status

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

ID	R/W	Field	Value ID	Value	Description
A	R	PDUSTAT			Status on payload length vs. PCNF1.MAXLEN
			LessThan	0	Payload less than PCNF1.MAXLEN
			GreaterThan	1	Payload greater than PCNF1.MAXLEN
B	R	CISTAT			Status on what rate packet is received with in Long Range
			LR125kbit	0	Frame is received at 125 kbps
			LR500kbit	1	Frame is received at 500 kbps

7.26.15.83 CTESTATUS

Address offset: 0x44C

CTEInfo parsed from received packet

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

ID	R/W	Field	Value ID	Value	Description
A	R	CTETIME			CTETime parsed from packet
B	R	RFU			RFU parsed from packet
C	R	CTETYPE			CTEType parsed from packet

7.26.15.84 DFESTATUS

Address offset: 0x458

DFE status information

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

ID	R/W	Field	Value ID	Value	Description
A	R	SWITCHINGSTATE			Internal state of switching state machine
			Idle	0	Switching state Idle
			Offset	1	Switching state Offset
			Guard	2	Switching state Guard
			Ref	3	Switching state Ref
			Switching	4	Switching state Switching
			Ending	5	Switching state Ending
B	R	SAMPLINGSTATE			Internal state of sampling state machine
			Idle	0	Sampling state Idle
			Sampling	1	Sampling state Sampling

7.26.15.85 PACKETPTR

Address offset: 0x504

Packet pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x01000000				**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 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	PACKETPTR			Packet pointer Packet address to be used for the next transmission or reception. When transmitting, the packet pointed to by this address will be transmitted and when receiving, the received packet will be written to this address. This address is a byte aligned RAM address. See the memory chapter for details about which memories are avilable for EasyDMA.

7.26.15.86 FREQUENCY

Address offset: 0x508

Frequency

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**

ID	R/W	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 and 2500 MHz Frequency = 2400 + FREQUENCY (MHz)
			Low	1	Channel map between 2360 MHz and 2460 MHz Frequency = 2360 + FREQUENCY (MHz)

7.26.15.87 TXPOWER

Address offset: 0x50C

Output power

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

ID	R/W	Field	Value ID	Value	Description	
A	RW	TXPOWER			RADIO output power Output power in number of dBm, i.e. if the value -20 is specified the output power will be set to -20 dBm. When the radio is operated on 1.2 V, the output power is increased by 3 dBm, i.e. if the value -20 is specified, the output power will be set to -17 dBm. See VREQCTRL — Voltage request control on page 60 for how to control voltage.	
			0dBm	0x0	0 dBm	
			Neg1dBm	0xFF	-1 dBm	
			Neg2dBm	0xFE	-2 dBm	
			Neg3dBm	0xFD	-3 dBm	
			Neg4dBm	0xFC	-4 dBm	
			Neg5dBm	0xFB	-5 dBm	
			Neg6dBm	0xFA	-6 dBm	
			Neg7dBm	0xF9	-7 dBm	
			Neg8dBm	0xF8	-8 dBm	
			Neg12dBm	0xF4	-12 dBm	
			Neg16dBm	0xF0	-16 dBm	
			Neg20dBm	0xEC	-20 dBm	
			Neg30dBm	0xE2	-40 dBm	Deprecated
			Neg40dBm	0xD8	-40 dBm	

7.26.15.88 MODE

Address offset: 0x510

Data rate and modulation

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MODE			Radio data rate and modulation setting. The radio supports frequency-shift keying (FSK) modulation.
			Nrf_1Mbit	0	1 Mbps Nordic proprietary radio mode
			Nrf_2Mbit	1	2 Mbps Nordic proprietary radio mode
			Ble_1Mbit	3	1 Mbps BLE
			Ble_2Mbit	4	2 Mbps BLE
			Ble_LR125Kbit	5	Long Range 125 kbps TX, 125 kbps and 500 kbps RX
			Ble_LR500Kbit	6	Long Range 500 kbps TX, 125 kbps and 500 kbps RX
			Ieee802154_250Kbit	15	IEEE 802.15.4-2006 250 kbps

7.26.15.89 PCNF0

Address offset: 0x514

Packet configuration register 0

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LFLEN			Length on air of LENGTH field in number of bits
C	RW	S0LEN			Length on air of S0 field in number of bytes
E	RW	S1LEN			Length on air of S1 field in number of bits
F	RW	S1INCL			Include or exclude S1 field in RAM
			Automatic	0	Include S1 field in RAM only if S1LEN > 0
			Include	1	Always include S1 field in RAM independent of S1LEN
G	RW	CILEN			Length of code indicator - Long Range
H	RW	PLEN			Length of preamble on air. Decision point: TASKS_START task
			8bit	0	8-bit preamble
			16bit	1	16-bit preamble
			32bitZero	2	32-bit zero preamble - used for IEEE 802.15.4
			LongRange	3	Preamble - used for Bluetooth LE Long Range
I	RW	CRCINC			Indicates if LENGTH field contains CRC or not
			Exclude	0	LENGTH does not contain CRC
			Include	1	LENGTH includes CRC
J	RW	TERMLEN			Length of TERM field in Long Range operation

7.26.15.90 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 B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXLEN		[0..255]	Maximum length of packet payload. If the packet payload is larger than MAXLEN, the radio will truncate the payload to MAXLEN.
B	RW	STATLEN		[0..255]	Static length in number of bytes The static length parameter is added to the total length of the payload when sending and receiving packets, e.g. if the static length is set to N the radio will receive or send N bytes more than what is defined in the LENGTH field of the packet.
C	RW	BALEN		[2..4]	Base address length in number of bytes The address field is composed of the base address and the one byte long address prefix, e.g. set BALEN=2 to get a total address of 3 bytes.
D	RW	ENDIAN			On-air endianness of packet, this applies to the S0, LENGTH, S1, and the PAYLOAD fields.
			Little	0	Least significant bit on air first
			Big	1	Most significant bit on air first
E	RW	WHITEEN			Enable or disable packet whitening Including the address field to CRC check is not supported for whitened packets.
			Disabled	0	Disable
			Enabled	1	Enable

7.26.15.91 BASE0

Address offset: 0x51C

Base address 0

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

ID	R/W	Field	Value ID	Value	Description
A	RW	BASE0			Base address 0

7.26.15.92 BASE1

Address offset: 0x520

Base address 1

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

ID	R/W	Field	Value ID	Value	Description
A	RW	BASE1			Base address 1

7.26.15.93 PREFIX0

Address offset: 0x524

Prefixes bytes for logical addresses 0-3

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-D	RW	AP[i] (i=0..3)			Address prefix i.

7.26.15.94 PREFIX1

Address offset: 0x528

Prefixes bytes for logical addresses 4-7

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-D	RW	AP[i] (i=4..7)			Address prefix i.

7.26.15.95 TXADDRESS

Address offset: 0x52C

Transmit address select

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

7.26.15.96 RXADDRESSES

Address offset: 0x530

Receive address select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	ADDR[i] (i=0..7)			Enable or disable reception on logical address i.
			Disabled	0	Disable
			Enabled	1	Enable

7.26.15.97 CRCCNF

Address offset: 0x534

CRC configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LEN		[1..3]	CRC length in number of bytes For MODE Ble_LR125Kbit and Ble_LR500Kbit, only LEN set to 3 is supported
			Disabled	0	CRC length is zero and CRC calculation is disabled
			One	1	CRC length is one byte and CRC calculation is enabled
			Two	2	CRC length is two bytes and CRC calculation is enabled
			Three	3	CRC length is three bytes and CRC calculation is enabled
B	RW	SKIPADDR			Include or exclude packet address field out of CRC calculation.
			Include	0	CRC calculation includes address field
			Skip	1	CRC calculation does not include address field. The CRC calculation will start at the first byte after the address.
			Ieee802154	2	CRC calculation as per 802.15.4 standard. Starting at first byte after length field.

7.26.15.98 CRCPOLY

Address offset: 0x538

CRC polynomial

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	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 hardwired internally to 1, and bit number 0 of the register content is ignored by the hardware. The following example is for an 8 bit CRC polynomial: x8 + x7 + x3 + x2 + 1 = 1 1000 1101 .

7.26.15.99 CRCINIT

Address offset: 0x53C

CRC initial value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CRCINIT			CRC initial value Initial value for CRC calculation

7.26.15.100 TIFS

Address offset: 0x544

Interframe spacing in µs

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TIFS			Interframe spacing in µs. Interframe space is the time interval between two consecutive packets. It is defined as the time, in microseconds, from the end of the last bit of the previous packet to the start of the first bit of the subsequent packet.

7.26.15.101 RSSISAMPLE

Address offset: 0x548

RSSI sample

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	RSSISAMPLE		[0..127]	RSSI sample. RSSI sample result. The value of this register is read as a positive value while the actual received signal strength is a negative value. Actual received signal strength is therefore as follows: received signal strength = -A dBm.

7.26.15.102 STATE

Address offset: 0x550

Current radio state

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	STATE			Current radio state
			Disabled	0	RADIO is in the Disabled state
			RxRu	1	RADIO is in the RXRU state
			RxIdle	2	RADIO is in the RXIDLE state
			Rx	3	RADIO is in the RX state
			RxDisable	4	RADIO is in the RXDISABLED state
			TxRu	9	RADIO is in the TXRU state
			TxIdle	10	RADIO is in the TXIDLE state
			Tx	11	RADIO is in the TX state
			TxDisable	12	RADIO is in the TXDISABLED state

7.26.15.103 DATAWHITEIV

Address offset: 0x554

Data whitening initial value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000040				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DATAWHITEIV			Data whitening initial value. Bit 6 is hardwired 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.

7.26.15.104 BCC

Address offset: 0x560

Bit counter compare

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

7.26.15.105 DAB[n] (n=0..7)

Address offset: 0x600 + (n × 0x4)

Device address base segment n

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

7.26.15.106 DAP[n] (n=0..7)

Address offset: 0x620 + (n × 0x4)

Device address prefix n

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

7.26.15.107 DACNF

Address offset: 0x640

Device address match configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	ENA[i] (i=0..7)			Enable or disable device address matching using device address i
			Disabled	0	Disabled
			Enabled	1	Enabled
I-P	RW	TXADD[i] (i=0..7)			TxAdd for device address i

7.26.15.108 MHRMATCHCONF

Address offset: 0x644

Search pattern configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	MHRMATCHCONF			Search pattern configuration

7.26.15.109 MHRMATCHMAS

Address offset: 0x648

Pattern mask

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	MHRMATCHMAS			Pattern mask

7.26.15.110 MODECNF0

Address offset: 0x650

Radio mode configuration register 0

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C A
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RU			Radio ramp-up time
			Default	0	Default ramp-up time (tRXEN and tTXEN), compatible with firmware written for nRF51
			Fast	1	Fast ramp-up (tRXEN,FAST and tTXEN,FAST), see electrical specifications for more information When enabled, TIFS is not enforced by hardware and software needs to control when to turn on the Radio

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C A
Reset 0x00000200				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
C	RW	DTX			Default TX value Specifies what the RADIO will transmit when it is not started, i.e. between: RADIO.EVENTS_READY and RADIO.TASKS_START RADIO.EVENTS_END and RADIO.TASKS_START RADIO.EVENTS_END and RADIO.EVENTS_DISABLED For IEEE 802.15.4 250 kbps mode only Center is a valid setting For Bluetooth Low Energy Long Range mode only Center is a valid setting
			B1	0	Transmit '1'
			B0	1	Transmit '0'
			Center	2	Transmit center frequency When tuning the crystal for center frequency, the RADIO must be set in DTX = Center mode to be able to achieve the expected accuracy

7.26.15.111 SFD

Address offset: 0x660

IEEE 802.15.4 start of frame delimiter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x000000A7				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	SFD			IEEE 802.15.4 start of frame delimiter

7.26.15.112 EDCNT

Address offset: 0x664

IEEE 802.15.4 energy detect loop count

Number of iterations to perform an ED scan. If set to 0 one scan is performed, otherwise the specified number + 1 of ED scans will be performed and the max ED value tracked in EDSAMPLE.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	EDCNT			IEEE 802.15.4 energy detect loop count

7.26.15.113 EDSAMPLE

Address offset: 0x668

IEEE 802.15.4 energy detect level

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	EDLVL		[0..127]	IEEE 802.15.4 energy detect level Register value must be converted to IEEE 802.15.4 range by an 8-bit saturating multiplication by factor ED_RSSISCALE, as shown in the code example for ED sampling

7.26.15.114 CCACTRL

Address offset: 0x66C

IEEE 802.15.4 clear channel assessment control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A
Reset 0x052D0000				**0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CCAMODE			CCA mode of operation
			EdMode	0	Energy above threshold Will report busy whenever energy is detected above CCAEDTHRES
			CarrierMode	1	Carrier seen Will report busy whenever compliant IEEE 802.15.4 signal is seen
			CarrierAndEdMode	2	Energy above threshold AND carrier seen
			CarrierOrEdMode	3	Energy above threshold OR carrier seen
			EdModeTest1	4	Energy above threshold test mode that will abort when first ED measurement over threshold is seen. No averaging.
B	RW	CCAEDTHRES			CCA energy busy threshold. Used in all the CCA modes except CarrierMode. Must be converted from IEEE 802.15.4 range by dividing by factor ED_RSSISCALE - similar to EDSAMPLE register
C	RW	CCACORRTHRES			CCA correlator busy threshold. Only relevant to CarrierMode, CarrierAndEdMode, and CarrierOrEdMode.
D	RW	CCACORRCNT			Limit for occurances above CCACORRTHRES. When not equal to zero the corrolator based signal detect is enabled.

7.26.15.115 DFEMODE

Address offset: 0x900

Whether to use Angle-of-Arrival (AOA) or Angle-of-Departure (AOD)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DFEOPMODE			Direction finding operation mode
			Disabled	0	Direction finding mode disabled
			AoD	2	Direction finding mode set to AoD
			AoA	3	Direction finding mode set to AoA

7.26.15.116 CTEINLINECONF

Address offset: 0x904

Configuration for CTE inline mode

Bit number																																
ID																																
Reset 0x00002800																																
ID	R/W	Field	Value ID	Value	Description																											
A	RW	CTEINLINECTRLEN			Enable parsing of CTEInfo from received packet in BLE modes																											
			Enabled	1	Parsing of CTEInfo is enabled																											
			Disabled	0	Parsing of CTEInfo is disabled																											
B	RW	CTEINFOINS1			CTEInfo is S1 byte or not																											
			InS1	1	CTEInfo is in S1 byte (data PDU)																											
			NotInS1	0	CTEInfo is NOT in S1 byte (advertising PDU)																											
C	RW	CTEERRORHANDLING			Sampling/switching if CRC is not OK																											
			Yes	1	Sampling and antenna switching also when CRC is not OK																											
			No	0	No sampling and antenna switching when CRC is not OK																											
E	RW	CTETIMEVALIDRANGE			Max range of CTETime Valid range is 2-20 in the Bluetooth Core Specification. If larger than 20, it can be an indication of an error in the received packet.																											
			20	0	20 in 8 µs unit (default) Set to 20 if parsed CTETime is larger than 20																											
			31	1	31 in 8 µs unit																											
			63	2	63 in 8 µs unit																											
F	RW	CTEINLINERXMODE1US			Spacing between samples for the samples in the SWITCHING period when CTEINLINEMODE is set. AoD 1 µs																											
			4us	1	4 µs																											
			2us	2	2 µs																											
			1us	3	1 µs																											
			500ns	4	0.5 µs																											
			250ns	5	0.25 µs																											
			125ns	6	0.125 µs																											
G	RW	CTEINLINERXMODE2US			Spacing between samples for the samples in the SWITCHING period when CTEINLINEMODE is set. When the device is in AoD mode, this is used when the received CTEType is "AoD 2 µs". When in AoA mode, this is used when TSWITCHSPACING is 4 µs.																											
			4us	1	4 µs																											
			2us	2	2 µs																											
			1us	3	1 µs																											
			500ns	4	0.5 µs																											
			250ns	5	0.25 µs																											
			125ns	6	0.125 µs																											
H	RW	S0CONF			S0 bit pattern to match The least significant bit always corresponds to the first bit of S0 received.																											

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I I I I I I I I H H H H H H H H G G G F F F E E C B A
Reset 0x00002800				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
I	RW	S0MASK			S0 bit mask to set which bit to match The least significant bit always corresponds to the first bit of S0 received.

7.26.15.117 DFECTRL1

Address offset: 0x910

Various configuration for Direction finding

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I I I I H H H H G G G F E E E C C C B A A A A A A
Reset 0x00023282				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	NUMBEROF8US			Length of the AoA/AoD procedure in number of 8 µs units Always used in TX mode, but in RX mode only when CTEINLINECTRLEN is 0
B	RW	DFEINEXTENSION			Add CTE extension and do antenna switching/sampling in this extension
			CRC	1	AoA/AoD procedure triggered at end of CRC
			Payload	0	Antenna switching/sampling is done in the packet payload
C	RW	TSWITCHSPACING			Interval between every time the antenna is changed in the SWITCHING state
			4us	1	4 µs
			2us	2	2 µs
			1us	3	1 µs
E	RW	TSAMPLESPACINGREF			Interval between samples in the REFERENCE period
			4us	1	4 µs
			2us	2	2 µs
			1us	3	1 µs
			500ns	4	0.5 µs
			250ns	5	0.25 µs
			125ns	6	0.125 µs
F	RW	SAMPLETYPE			Whether to sample I/Q or magnitude/phase
			IQ	0	Complex samples in I and Q
			MagPhase	1	Complex samples as magnitude and phase
G	RW	TSAMPLESPACING			Interval between samples in the SWITCHING period when CTEINLINECTRLEN is 0 When CTEINLINECTRLEN is 1, CTEINLINERXMODE1US or CTEINLINERXMODE2US is used instead of TSAMPLESPACING.
			4us	1	4 µs
			2us	2	2 µs
			1us	3	1 µs
			500ns	4	0.5 µs
			250ns	5	0.25 µs
			125ns	6	0.125 µs
H	RW	REPEATPATTERN			Repeat each individual antenna pattern N times sequentially, i.e. P0, P0, P1, P1, P2, P2, P3, P3, etc.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I I I I H H H H G G G F E E E C C C B A A A A A A
Reset 0x00023282				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
			NoRepeat	0	Do not repeat (1 time in total)
I	RW	AGCBACKOFFGAIN			Gain will be lowered by the specified number of gain steps at the start of CTE At the start of receiving the CTE, the gain is lowered by the specified number of gain steps (approximately 3 dB ± 1 dB per step). The initial gain reduction is implemented in the analog front-end, thus the gain back-off can be used to prevent sample saturation.

7.26.15.118 DFECTRL2

Address offset: 0x914

Start offset for Direction finding

Bit number				31 30 29 28 27 26 25 24 23 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 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	R/W	Field	Value ID	Value	Description
A	RW	TSWITCHOFFSET			Signed value offset after the end of the CRC before starting switching in number of 16 MHz clock cycles Decreasing TSWITCHOFFSET beyond the trigger of the AoA/AoD procedure will have no effect
B	RW	TSAMPLEOFFSET			Signed value offset in number of 16 MHz clock cycles for fine tuning of the sampling instant for all IQ samples. With TSAMPLEOFFSET=0 the first sample is taken immediately at the start of the reference period Decreasing TSAMPLEOFFSET beyond the trigger of the AoA/AoD procedure will have no effect

7.26.15.119 SWITCHPATTERN

Address offset: 0x928

GPIO patterns to be used for each antenna

Maximum 8 GPIOs can be controlled. To secure correct signal levels on the pins, the pins must be configured in the GPIO peripheral as described in Pin configuration.

If, during switching, the total number of antenna slots is bigger than the number of written patterns, the RADIO loops back to the pattern used after the reference pattern.

A minimum number of three patterns must be written.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SWITCHPATTERN			Fill array of GPIO patterns for antenna control. The GPIO pattern array size is 40 entries. When written, bit n corresponds to the GPIO configured in PSEL.DFEGPIO[n]. When read, returns the number of GPIO patterns written since the last time the array was cleared. Use CLEARPATTERN to clear the array.

7.26.15.120 CLEARPATTERN

Address offset: 0x92C

Clear the GPIO pattern array for antenna control

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CLEARPATTERN			Clears GPIO pattern array for antenna control
			Clear	1	Clear the GPIO pattern

7.26.15.121 PSEL.DFEGPIO[n] (n=0..7)

Address offset: 0x930 + (n × 0x4)

Pin select for DFE pin n

Must be set before enabling the radio

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

7.26.15.122 DFEPACKET.PTR

Address offset: 0x950

Data pointer

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x01000000	**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**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.26.15.123 DFEPACKET.MAXCNT

Address offset: 0x954

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
Reset 0x00001000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT			Maximum number of buffer words to transfer

7.26.15.124 DFEPACKET.AMOUNT

Address offset: 0x958

Number of samples 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																																
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	R	AMOUNT			Number of samples transferred in the last transaction

7.26.15.125 POWER

Address offset: 0xFFC

Peripheral power control

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

ID	R/W	Field	Value ID	Value	Description
A	RW	POWER			Peripheral power control. The peripheral and its registers will be reset to its initial state by switching the peripheral off and then back on again.
			Disabled	0	Peripheral is powered off
			Enabled	1	Peripheral is powered on

7.26.16 Electrical specification

7.26.16.1 General radio characteristics

Symbol	Description	Min.	Typ.	Max.	Units
f_{OP}	Operating frequencies	2360		2500	MHz
$f_{PLL,CH,SP}$	PLL channel spacing		1.0		MHz
$f_{DELTA,1M}$	Frequency deviation @ 1 Mbps		±170		kHz
$f_{DELTA,BLE,1M}$	Frequency deviation @ Bluetooth LE 1 Mbps		±250		kHz
$f_{DELTA,2M}$	Frequency deviation @ 2 Mbps		±320		kHz
$f_{DELTA,BLE,2M}$	Frequency deviation @ Bluetooth LE 2 Mbps		±500		kHz
fsk_{BPS}	On-the-air data rate	125		2000	kbps
$f_{chip,\ IEEE\ 802.15.4}$	Chip rate in IEEE 802.15.4 mode		2000		kchip/s

7.26.16.2 Radio current consumption (transmitter)

Symbol	Description	Min.	Typ.	Max.	Units
$I_{TX,0dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = 0 dBm		3.4		mA
$I_{TX,0dBM}$	TX only run current P_{RF} = 0 dBm		9.1		mA
$I_{TX,MINUS4dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -4 dBm		2.7		mA
$I_{TX,MINUS4dBM}$	TX only run current P_{RF} = -4 dBm		7.2		mA
$I_{TX,MINUS8dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -8 dBm		2.2		mA
$I_{TX,MINUS8dBM}$	TX only run current P_{RF} = -8 dBm		5.8		mA
$I_{TX,MINUS12dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -12 dBm		2.0		mA
$I_{TX,MINUS12dBM}$	TX only run current P_{RF} = -12 dBm		5.0		mA
$I_{TX,MINUS16dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -16 dBm		1.8		mA
$I_{TX,MINUS16dBM}$	TX only run current P_{RF} = -16 dBm		4.5		mA
$I_{TX,MINUS20dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -20 dBm		1.7		mA
$I_{TX,MINUS20dBM}$	TX only run current P_{RF} = -20 dBm		4.2		mA
$I_{TX,MINUS40dBM,DCDC}$	TX only run current DC/DC, 3 V, P_{RF} = -40 dBm		1.5		mA
$I_{TX,MINUS40dBM}$	TX only run current P_{RF} = -40 dBm		3.8		mA
$I_{START,TX,DCDC}$	TX start-up current DC/DC, 3 V, P_{RF} = 4 dBm		2.4		mA
$I_{START,TX}$	TX start-up current, P_{RF} = 4 dBm		5.4		mA

7.26.16.3 Radio current consumption (Receiver)

Symbol	Description	Min.	Typ.	Max.	Units
$I_{RX,1M,DCDC}$	RX only run current DC/DC, 3 V, 1 Mbps/1 Mbps Bluetooth LE mode		2.7		mA
$I_{RX,1M}$	RX only run current LDO, 3 V, 1 Mbps/1 Mbps Bluetooth LE mode		6.7		mA
$I_{RX,2M,DCDC}$	RX only run current DC/DC, 3 V, 2 Mbps/2 Mbps Bluetooth LE mode		3.1		mA
$I_{RX,2M}$	RX only run current LDO, 3 V, 2 Mbps/2 Mbps Bluetooth LE mode		7.9		mA
$I_{START,RX,1M,DCDC}$	RX start-up current DC/DC, 3 V, 1 Mbps/1 Mbps Bluetooth LE mode	..	..	..	mA
$I_{START,RX,1M}$	RX start-up current 1 Mbps/1 Mbps Bluetooth LE mode	..	..	..	mA

7.26.16.4 Transmitter specification

Symbol	Description	Min.	Typ.	Max.	Units
P_{RF}	Maximum output power		3.0		dBm
P_{RFC}	RF power control range		23.0		dB
P_{RFCR}	RF power accuracy		±2		dB
$P_{RF1,1}$	1st Adjacent Channel Transmit Power 1 MHz (1 Mbps)		-24		dBc
$P_{RF2,1}$	2nd Adjacent Channel Transmit Power 2 MHz (1 Mbps)		-52		dBc
$P_{RF1,2}$	1st Adjacent Channel Transmit Power 2 MHz (2 Mbps)		-25		dBc
$P_{RF2,2}$	2nd Adjacent Channel Transmit Power 4 MHz (2 Mbps)		-50		dBc
E_{vm}	Error vector magnitude in IEEE 802.15.4 mode	..	..	..	%rms
$P_{harm2nd, IEEE 802.15.4}$	2nd harmonics in IEEE 802.15.4 mode		-51		dBm
$P_{harm3rd, IEEE 802.15.4}$	3rd harmonics in IEEE 802.15.4 mode		-51		dBm

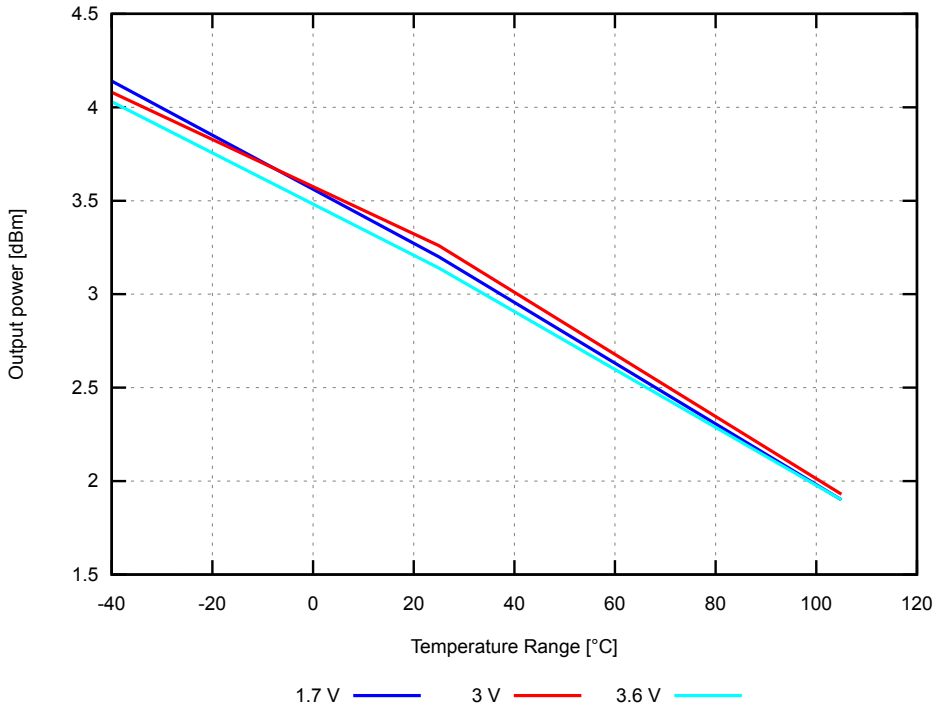

Figure 170: Output power, 1 Mbps Bluetooth low energy mode, at maximum TXPOWER setting (typical values)

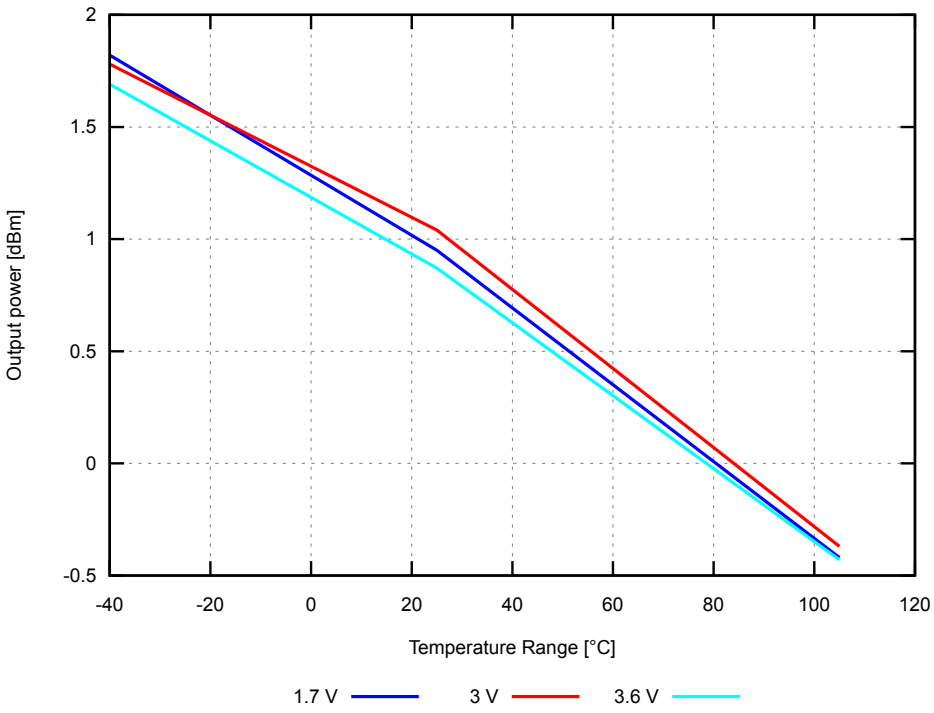

Figure 171: Output power, 1 Mbps Bluetooth low energy mode, at 0 dBm TXPOWER setting (typical values)

7.26.16.5 Receiver operation

Symbol	Description	Min.	Typ.	Max.	Units
$P_{RX,MAX}$	Maximum received signal strength at < 0.1% PER		0		dBm
$P_{SENS,IT,1M}$	Sensitivity, 1 Mbps nRF mode ideal transmitter [12]		-95		dBm
$P_{SENS,IT,2M}$	Sensitivity, 2 Mbps nRF mode ideal transmitter [12]		-92		dBm
$P_{SENS,IT,SP,1M,BLE}$	Sensitivity, 1 Mbps Bluetooth LE ideal transmitter, packet length ≤ 37 bytes BER = 1E-3[13]		-98		dBm
$P_{SENS,IT,LP,1M,BLE}$	Sensitivity, 1 Mbps Bluetooth LE ideal transmitter, packet length ≥ 128 bytes BER = 1E-4 [14]		-97		dBm
$P_{SENS,IT,SP,2M,BLE}$	Sensitivity, 2 Mbps Bluetooth LE ideal transmitter, packet length ≤ 37 bytes		-95		dBm
$P_{SENS,IT,BLE\ LE125k}$	Sensitivity, 125 kbps Bluetooth LE mode		-104		dBm
$P_{SENS,IT,BLE\ LE500k}$	Sensitivity, 500 kbps Bluetooth LE mode		-100		dBm
$P_{SENS,IEEE\ 802.15.4}$	Sensitivity in IEEE 802.15.4 mode		-101		dBm

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

[13] As defined in the *Bluetooth Core Specification v4.0 Volume 6: Core System Package (Low Energy Controller Volume)*.

[14] Equivalent BER limit < 10E-04.

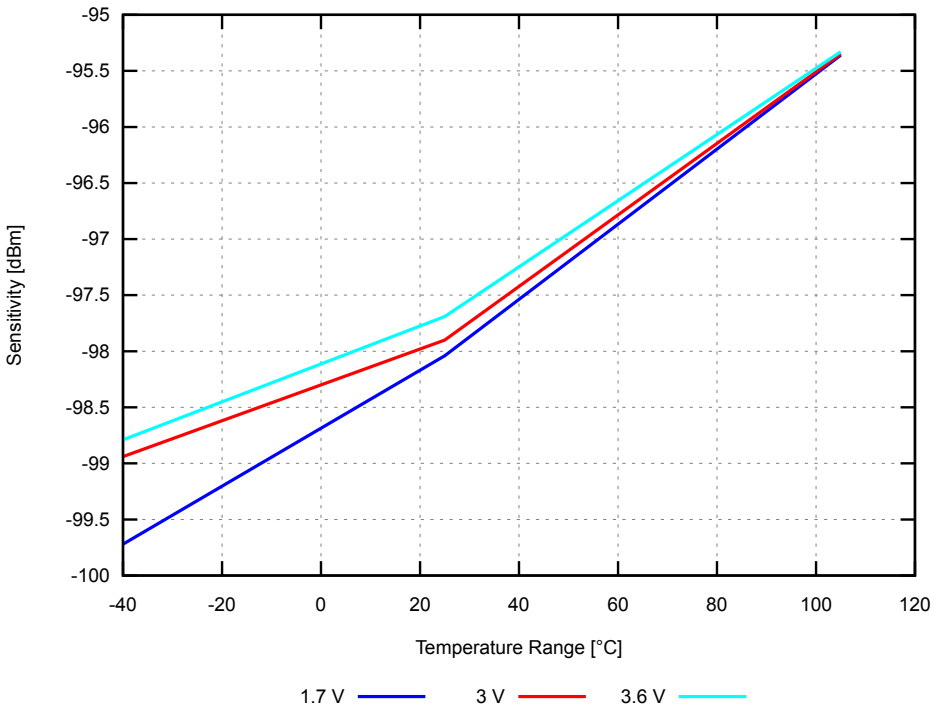

Figure 172: Sensitivity, 1 Mbps Bluetooth low energy mode, Regulator = DCDC (typical values)

7.26.16.6 RX selectivity

RX selectivity with equal modulation on interfering signal[15]

Symbol	Description	Min.	Typ.	Max.	Units
$C/I_{1M,co\text{-}channel}$	1Mbps mode, co-channel interference	..	..	..	dB
$C/I_{1M,-1MHz}$	1 Mbps mode, Adjacent (-1 MHz) interference	..	..	..	dB
$C/I_{1M,+1MHz}$	1 Mbps mode, Adjacent (+1 MHz) interference	..	..	..	dB
$C/I_{1M,-2MHz}$	1 Mbps mode, Adjacent (-2 MHz) interference	..	..	..	dB
$C/I_{1M,+2MHz}$	1 Mbps mode, Adjacent (+2 MHz) interference	..	..	..	dB
$C/I_{1M,-3MHz}$	1 Mbps mode, Adjacent (-3 MHz) interference	..	..	..	dB
$C/I_{1M,+3MHz}$	1 Mbps mode, Adjacent (+3 MHz) interference	..	..	..	dB
$C/I_{1M,\pm6MHz}$	1 Mbps mode, Adjacent (≥6 MHz) interference	..	..	..	dB
$C/I_{1MBLE,co\text{-}channel}$	1 Mbps Bluetooth LE mode, co-channel interference		6.9		dB
$C/I_{1MBLE,-1MHz}$	1 Mbps Bluetooth LE mode, Adjacent (-1 MHz) interference		-2.6		dB
$C/I_{1MBLE,+1MHz}$	1 Mbps Bluetooth LE mode, Adjacent (+1 MHz) interference		-8.5		dB
$C/I_{1MBLE,-2MHz}$	1 Mbps Bluetooth LE mode, Adjacent (-2 MHz) interference		-27		dB
$C/I_{1MBLE,+2MHz}$	1 Mbps Bluetooth LE mode, Adjacent (+2 MHz) interference		-45		dB
$C/I_{1MBLE,>3MHz}$	1 Mbps Bluetooth LE mode, Adjacent (≥3 MHz) interference		-50		dB
$C/I_{1MBLE,image}$	Image frequency interference		-27		dB
$C/I_{1MBLE,image,1MHz}$	Adjacent (1 MHz) interference to in-band image frequency		-41		dB
$C/I_{2M,co\text{-}channel}$	2 Mbps mode, co-channel interference	..	..	..	dB
$C/I_{2M,-2MHz}$	2 Mbps mode, Adjacent (-2 MHz) interference	..	..	..	dB
$C/I_{2M,+2MHz}$	2 Mbps mode, Adjacent (+2 MHz) interference	..	..	..	dB
$C/I_{2M,-4MHz}$	2 Mbps mode, Adjacent (-4 MHz) interference	..	..	..	dB
$C/I_{2M,+4MHz}$	2 Mbps mode, Adjacent (+4 MHz) interference	..	..	..	dB

[15] Desired signal level at P_{IN} = -67 dBm. One interferer is used, having equal modulation as the desired signal. The input power of the interferer where the sensitivity equals BER = 1E-4 is presented.

Symbol	Description	Min.	Typ.	Max.	Units
$C/I_{2M,-6MHz}$	2 Mbps mode, Adjacent (-6 MHz) interference	..	..	..	dB
$C/I_{2M,+6MHz}$	2 Mbps mode, Adjacent (+6 MHz) interference	..	..	..	dB
$C/I_{2M,\geq12MHz}$	2 Mbps mode, Adjacent (≥12 MHz) interference	..	..	..	dB
$C/I_{2MBLE,co\text{-}channel}$	2 Mbps Bluetooth LE mode, co-channel interference		7.1		dB
$C/I_{2MBLE,-2MHz}$	2 Mbps Bluetooth LE mode, Adjacent (-2 MHz) interference		-2		dB
$C/I_{2MBLE,+2MHz}$	2 Mbps Bluetooth LE mode, Adjacent (+2 MHz) interference		-11		dB
$C/I_{2MBLE,-4MHz}$	2 Mbps Bluetooth LE mode, Adjacent (-4 MHz) interference		-22		dB
$C/I_{2MBLE,+4MHz}$	2 Mbps Bluetooth LE mode, Adjacent (+4 MHz) interference		-47		dB
$C/I_{2MBLE,\geq6MHz}$	2 Mbps Bluetooth LE mode, Adjacent (≥6 MHz) interference		-54		dB
$C/I_{2MBLE,image}$	Image frequency interference		-22		dB
$C/I_{2MBLE,image,\ 2MHz}$	Adjacent (2 MHz) interference to in-band image frequency		-42		dB
$C/I_{125k\ BLE\ LR,co\text{-}channel}$	125 kbps Bluetooth LE LR mode, co-channel interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,-1MHz}$	125 kbps Bluetooth LE LR mode, Adjacent (-1 MHz) interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,+1MHz}$	125 kbps Bluetooth LE LR mode, Adjacent (+1 MHz) interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,-2MHz}$	125 kbps Bluetooth LE LR mode, Adjacent (-2 MHz) interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,+2MHz}$	125 kbps Bluetooth LE LR mode, Adjacent (+2 MHz) interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,>3MHz}$	125 kbps Bluetooth LE LR mode, Adjacent (≥3 MHz) interference	..	..	..	dB
$C/I_{125k\ BLE\ LR,image}$	Image frequency interference	..	..	..	dB
$C/I_{IEEE\ 802.15.4,-5MHz}$	IEEE 802.15.4 mode, Adjacent (-5 MHz) rejection		-33		dB
$C/I_{IEEE\ 802.15.4,+5MHz}$	IEEE 802.15.4 mode, Adjacent (+5 MHz) rejection		-38		dB
$C/I_{IEEE\ 802.15.4,\ \pm10MHz}$	IEEE 802.15.4 mode, Alternate (±10 MHz) rejection		-50		dB

7.26.16.7 RX intermodulation

RX intermodulation. Desired signal level at P_{IN} = -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 desired signal. The input power of the interferers where the sensitivity equals BER = 1E-3 is presented.

Symbol	Description	Min.	Typ.	Max.	Units
$P_{IMD,5TH,1M}$	IMD performance, 1 Mbps, 5th offset channel, packet length ≤ 37 bytes	..	..	..	dBm
$P_{IMD,5TH,1M,BLE}$	IMD performance, Bluetooth LE 1 Mbps, 5th offset channel, packet length ≤ 37 bytes		-26		dBm
$P_{IMD,5TH,2M}$	IMD performance, 2 Mbps, 5th offset channel, packet length ≤ 37 bytes	..	..	..	dBm
$P_{IMD,5TH,2M,BLE}$	IMD performance, Bluetooth LE 2 Mbps, 5th offset channel, packet length ≤ 37 bytes		-22		dBm

7.26.16.8 Radio timing

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TXEN,BLE,1M}$	Time between TXEN task and READY event after channel FREQUENCY configured (1 Mbps Bluetooth LE and 150 µs TIFS)		140		µs

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TXEN,FAST,BLE,1M}$	Time between TXEN task and READY event after channel FREQUENCY configured (1 Mbps Bluetooth LE with fast ramp-up and 150 µs TIFS)		40		µs
$t_{TXDIS,BLE,1M}$	When in TX, delay between DISABLE task and DISABLED event for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		6		µs
$t_{RXEN,BLE,1M}$	Time between the RXEN task and READY event after channel FREQUENCY configured (1 Mbps Bluetooth LE)		140		µs
$t_{RXEN,FAST,BLE,1M}$	Time between the RXEN task and READY event after channel FREQUENCY configured (1 Mbps Bluetooth LE with fast ramp-up)		40		µs
$t_{RXDIS,BLE,1M}$	When in RX, delay between DISABLE task and DISABLED event for MODE = Nrf_1Mbit and MODE = Ble_1Mbit		0		µs
$t_{TXDIS,BLE,2M}$	When in TX, delay between DISABLE task and DISABLED event for MODE = Nrf_2Mbit and MODE = Ble_2Mbit		4		µs
$t_{RXDIS,BLE,2M}$	When in RX, delay between DISABLE task and DISABLED event for MODE = Nrf_2Mbit and MODE = Ble_2Mbit		0		µs
$t_{TXEN,IEEE\ 802.15.4}$	Time between TXEN task and READY event after channel FREQUENCY configured (IEEE 802.15.4 mode)		130		µs
$t_{TXEN,FAST,IEEE\ 802.15.4}$	Time between TXEN task and READY event after channel FREQUENCY configured (IEEE 802.15.4 mode with fast ramp-up)		40		µs
$t_{TXDIS,IEEE\ 802.15.4}$	When in TX, delay between DISABLE task and DISABLED event (IEEE 802.15.4 mode)		21		µs
$t_{RXEN,IEEE\ 802.15.4}$	Time between the RXEN task and READY event after channel FREQUENCY configured (IEEE 802.15.4 mode)		130		µs
$t_{RXEN,FAST,IEEE\ 802.15.4}$	Time between the RXEN task and READY event after channel FREQUENCY configured (IEEE 802.15.4 mode with fast ramp-up)		40		µs
$t_{RXDIS,IEEE\ 802.15.4}$	When in RX, delay between DISABLE task and DISABLED event (IEEE 802.15.4 mode)		0.5		µs
$t_{RX\text{-}to\text{-}TX\ turnaround}$	Maximum TX-to-RX or RX-to-TX turnaround time in IEEE 802.15.4 mode		40		µs

7.26.16.9 Received signal strength indicator (RSSI) specifications

Symbol	Description	Min.	Typ.	Max.	Units
$RSSI_{ACC}$	RSSI accuracy		±2		dB
$RSSI_{RESOLUTION}$	RSSI resolution		1		dB
$RSSI_{PERIOD}$	RSSI sampling time from RSSI_START task		0.25		µs
$RSSI_{SETTLE}$	RSSI settling time after signal level change		15		µs

7.26.16.10 Jitter

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

7.26.16.11 IEEE 802.15.4 mode energy detection constants

Symbol	Description	Min.	Typ.	Max.	Units
ED_RSSISCALE	Scaling value when converting between hardware-reported value and dBm		5		
ED_RSSIOFFS	Offset value when converting between hardware-reported value and dBm		-93		

7.27 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.

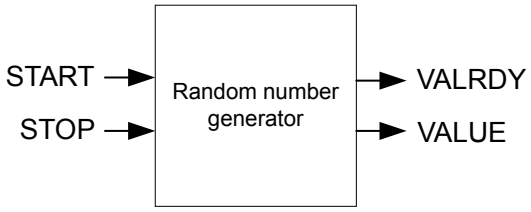

Figure 173: Random number generator

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.

7.27.1 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.

7.27.2 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.

7.27.3 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x41009000	NETWORK	RNG	RNG	NS	NA	Random number generator	

Table 132: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Task starting the random number generator
TASKS_STOP	0x004		Task stopping the random number generator
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP

Register	Offset	Security	Description
EVENTS_VALRDY	0x100		Event being generated for every new random number written to the VALUE register
PUBLISH_VALRDY	0x180		Publish configuration for event VALRDY
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
CONFIG	0x504		Configuration register
VALUE	0x508		Output random number

Table 133: Register overview

7.27.3.1 TASKS_START

Address offset: 0x000

Task starting the random number generator

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Task starting the random number generator
			Trigger	1	Trigger task

7.27.3.2 TASKS_STOP

Address offset: 0x004

Task stopping the random number generator

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Task stopping the random number generator
			Trigger	1	Trigger task

7.27.3.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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	B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.27.3.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.27.3.5 EVENTS_VALRDY

Address offset: 0x100

Event being generated for every new random number written to the VALUE register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_VALRDY			Event being generated for every new random number written to the VALUE register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.27.3.6 PUBLISH_VALRDY

Address offset: 0x180

Publish configuration for event VALRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event VALRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.27.3.7 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	VALRDY_STOP			Shortcut between event VALRDY and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.27.3.8 INTENSET

Address offset: 0x304

Enable interrupt

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

7.27.3.9 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.27.3.10 CONFIG

Address offset: 0x504

Configuration register

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DERCEN			Bias correction
			Disabled	0	Disabled
			Enabled	1	Enabled

7.27.3.11 VALUE

Address offset: 0x508

Output random number

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	VALUE		[0..255]	Generated random number

7.27.4 Electrical specification

7.27.4.1 RNG Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{RNG,START}$	Time from setting the START task to generation begins. This is a one-time delay on START signal and does not apply between samples.		128		µs
$t_{RNG,RAW}$	Run time per byte without bias correction. Uniform distribution of 0 and 1 is not guaranteed.		32		µs
$t_{RNG,BC}$	Run time per byte with bias correction. Uniform distribution of 0 and 1 is guaranteed. Time to generate a byte cannot be guaranteed.			122	µs

7.28 RTC — Real-time counter

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

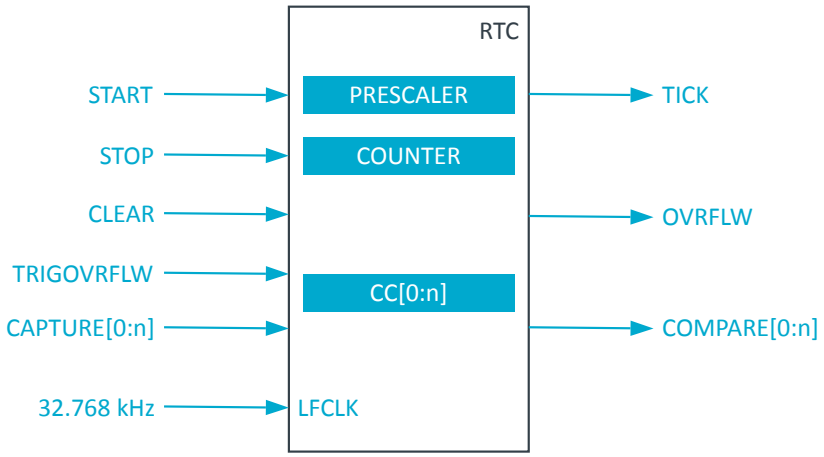

Figure 174: RTC block diagram

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

7.28.1 Clock source

The RTC will run off the LFCLK.

When started, the RTC will automatically request the LFCLK source with RC oscillator if the LFCLK is not already running.

See CLOCK — Clock control on page 69 for more information about clock sources.

7.28.2 Resolution versus overflow and the prescaler

The relationship between the prescaler, counter resolution, and overflow is summarized in the following table.

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

Table 134: RTC resolution versus overflow

The counter increment frequency is given by the following equation:

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

The PRESCALER register can only be written when the RTC is stopped.

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

Examples:

1. Desired COUNTER frequency 100 Hz (10 ms counter period)

   PRESCALER = round(32.768 kHz / 100 Hz) - 1 = 327

   f_{RTC} = 99.9 Hz

   10009.576 µs counter period
2. Desired COUNTER frequency 8 Hz (125 ms counter period)

   PRESCALER = round(32.768 kHz / 8 Hz) – 1 = 4095

   f_{RTC} = 8 Hz

   125 ms counter period

7.28.3 Counter register

The internal <<COUNTER>> register 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.

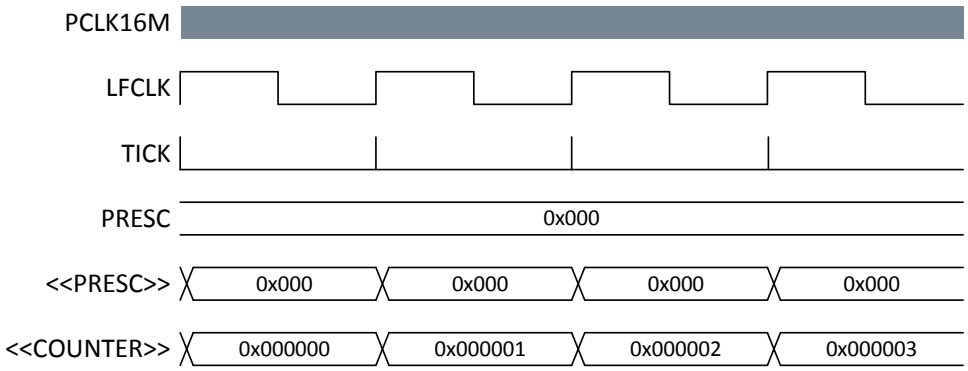

Figure 175: Timing diagram - COUNTER_PRESCALER_0

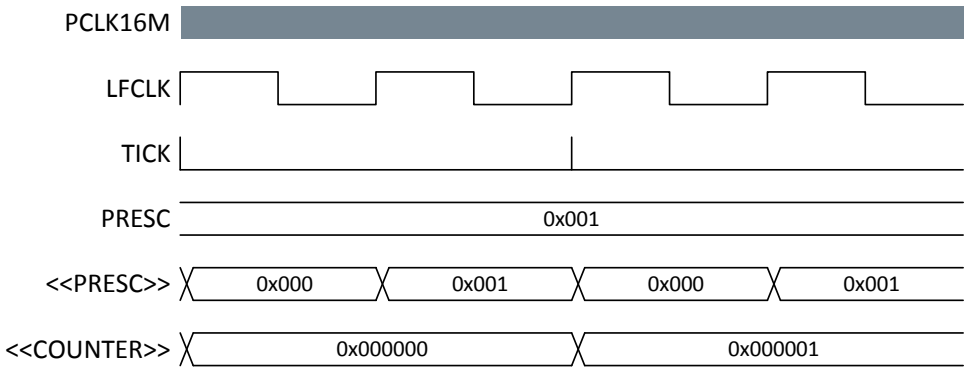

Figure 176: Timing diagram - COUNTER_PRESCALER_1

7.28.3.1 Reading the counter register

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

To ensure that the <<COUNTER>> is safely sampled (considering that an LFCLK transition may occur during a read), the CPU and core memory bus are halted for four PCLK16M cycles. In addition, the read takes the CPU two PCLK16M cycles, resulting in the COUNTER register read taking maximum six PCLK16M clock cycles.

7.28.4 Overflow

An OVRFLW event is generated on COUNTER register overflow (overflowing from `0xFFFFFF` to `0`).

The TRIGOVRFLW task will set the COUNTER value to `0xFFFFF0`, to allow software test of the overflow condition.

Note: The OVRFLW event is disabled by default.

7.28.5 Tick event

The TICK event enables low-power tickless RTOS implementation, as it optionally provides a regular interrupt source for an RTOS with no need for use of the ARM SysTick feature.

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

Note: The TICK event is disabled by default.

7.28.6 Event control

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

This means that the RTC implements a slightly different task and event system compared to the standard system described in Peripheral interface on page 145. The RTC task and event system is illustrated in the following figure.

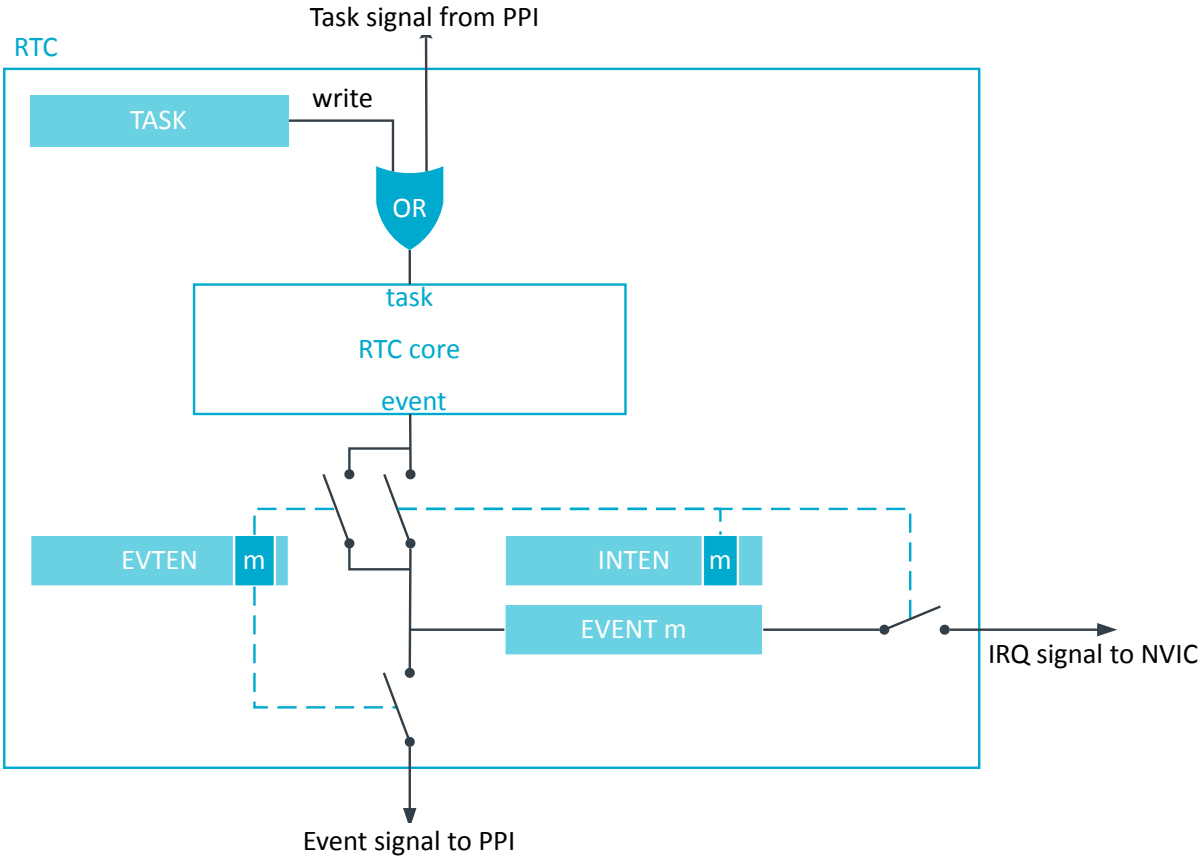

Figure 177: Tasks, events, and interrupts in the RTC

7.28.7 Capture

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

Every time TASKS_CAPTURE[n] is triggered, <<COUNTER>> is copied to the corresponding CC[n] register.

If the CAPTURE and CLEAR tasks are triggered at the same time, the CAPTURE task will be prioritized. This means that the CC[n] register for the corresponding CAPTURE[n] task will be set to the captured value before the counter is reset. There is a delay of 6 PCLK16M periods from when the TASKS_CAPTURE[n] is triggered until the corresponding CC[n] register is updated.

The CAPTURE[n] tasks will not generate COMPARE[n] events, even though CC[n] will then equal the COUNTER.

7.28.8 Compare

The RTC implements one COMPARE event for every available compare register.

When the COUNTER is incremented and then becomes equal to the value specified in the register CC[n], the corresponding compare event COMPARE[n] is generated.

When writing a CC[n] register, the RTC COMPARE event exhibits several behaviors. See the following figures for more information.

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

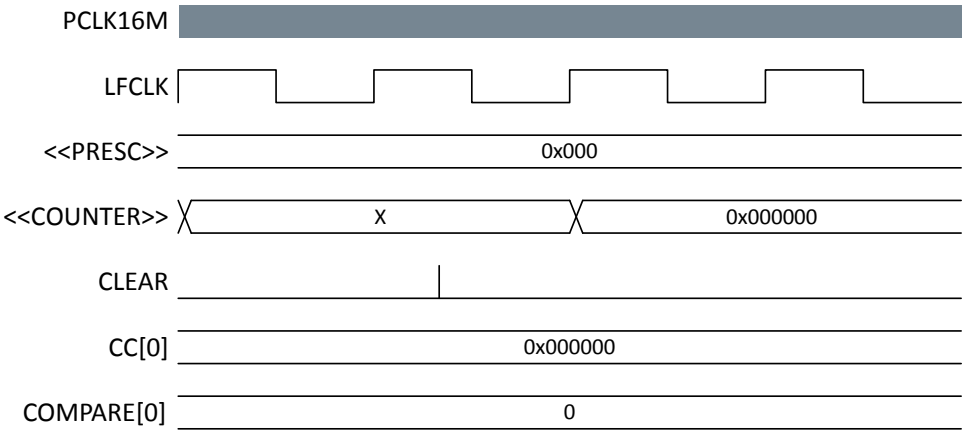

Figure 178: Timing diagram - COMPARE_CLEAR

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

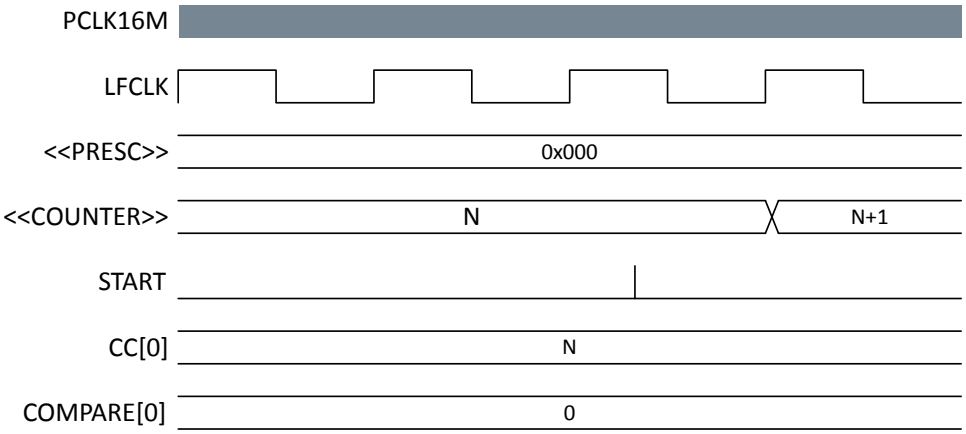

Figure 179: Timing diagram - COMPARE_START

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

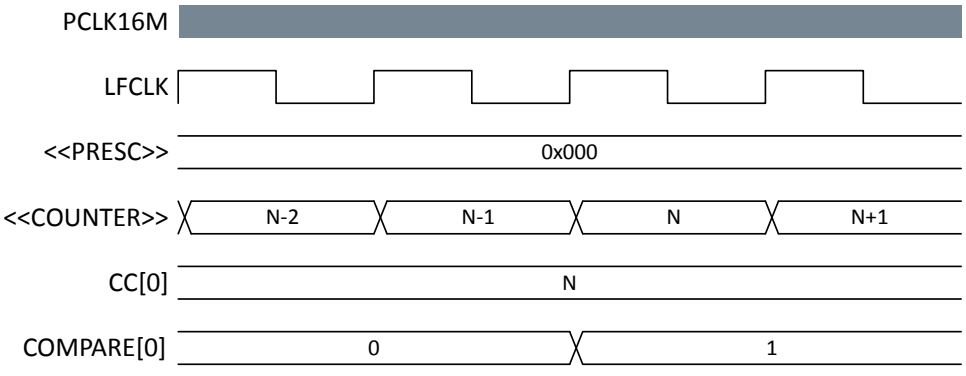

Figure 180: Timing diagram - COMPARE

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

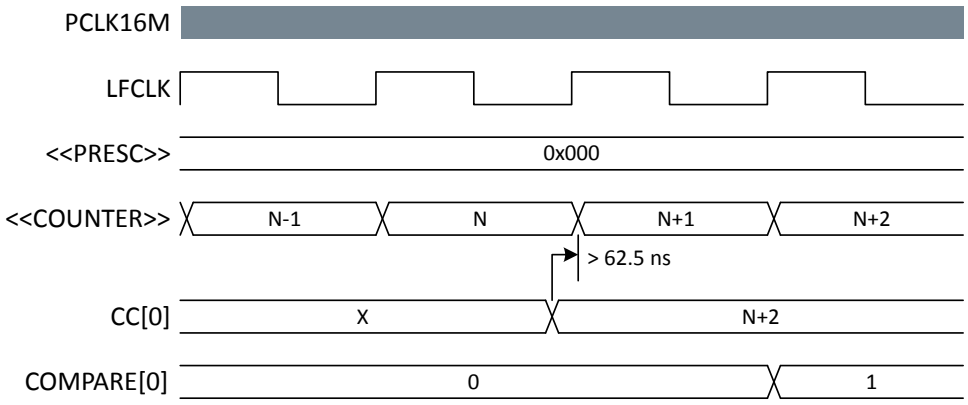

Figure 181: Timing diagram - COMPARE_N+2

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

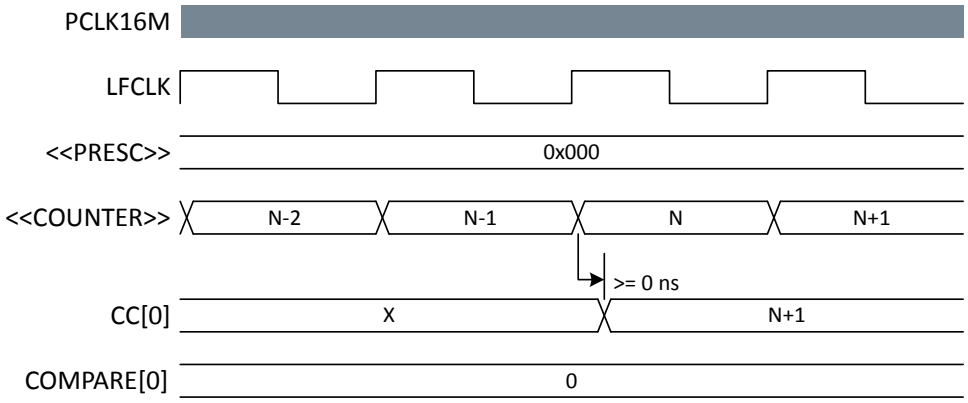

Figure 182: Timing diagram - COMPARE_N+1

If the COUNTER value is N, and the current CC 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.

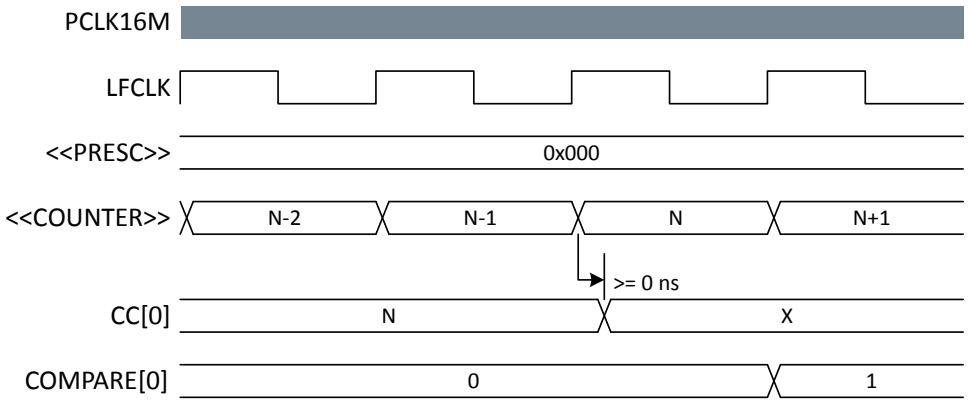

Figure 183: Timing diagram - COMPARE_N-1

If the COMPARE[i]_CLEAR short is enabled, the COUNTER will be cleared one LFClk after the COMPARE event.

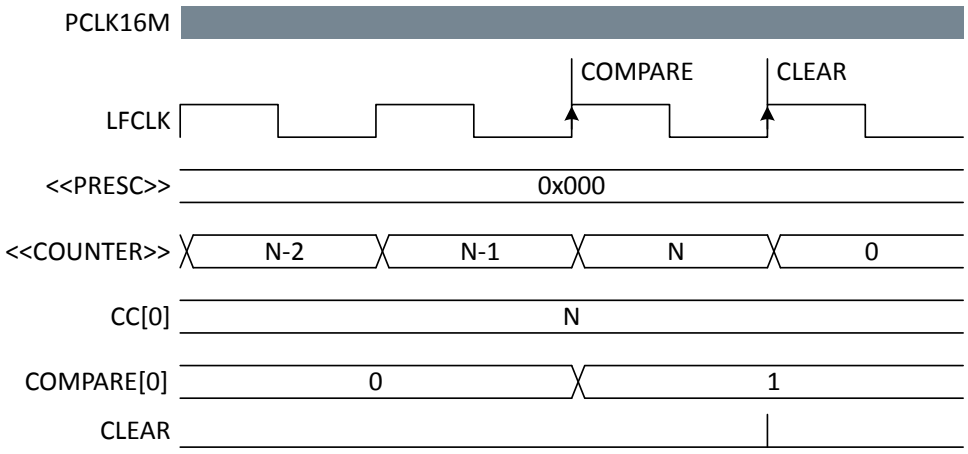

Figure 184: Timing diagram - COMPARE_CLEAR

7.28.9 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 that are 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 a read from an internal COUNTER register in the LFCLK domain. The COUNTER register is modified each time the RTC ticks. The registers are synchronised between the two clock domains (PCLK16M and LFCLK).

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

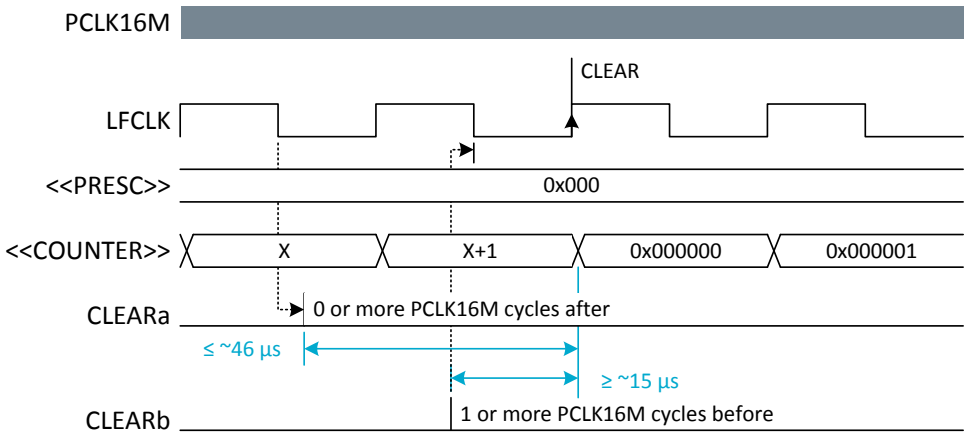

Figure 185: Timing diagram - DELAY_CLEAR

When a STOP task is triggered, the PCLK16M domain will immediately prevent the generation of any EVENTS from the RTC. However, as seen in the following figure, the COUNTER value can still increment one final time.

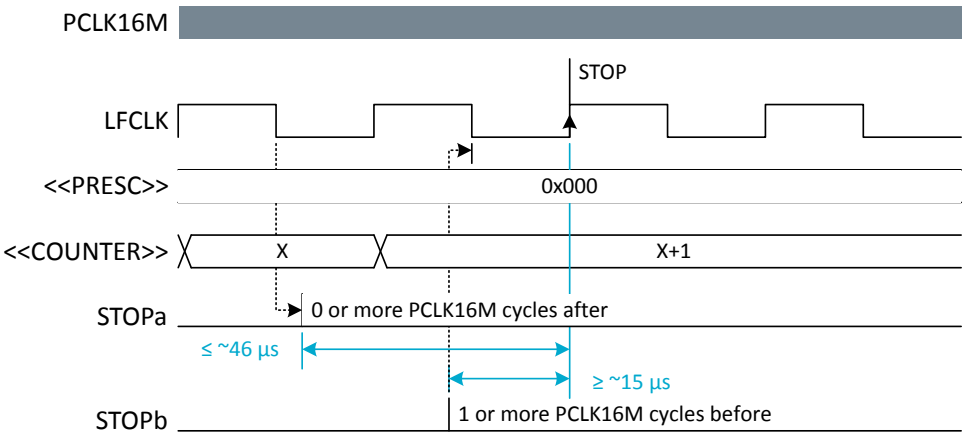

Figure 186: Timing diagram - DELAY_STOP

The START task will start the RTC. Assuming that the LFCLK was previously running and stable, the first increment of COUNTER (and instance of TICK event) will be typically after 30.5 µs ±15 µs. Additional delay will occur if the RTC is started before the LFCLK is running, see CLOCK — Clock control on page 69 for LFLK startup times. The software should therefore wait for the first TICK if it has to make sure that 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 also add additional delay as previously described. The figures show the smallest and largest delays on the START task, appearing as a ±15 µs jitter on the first COUNTER increment.

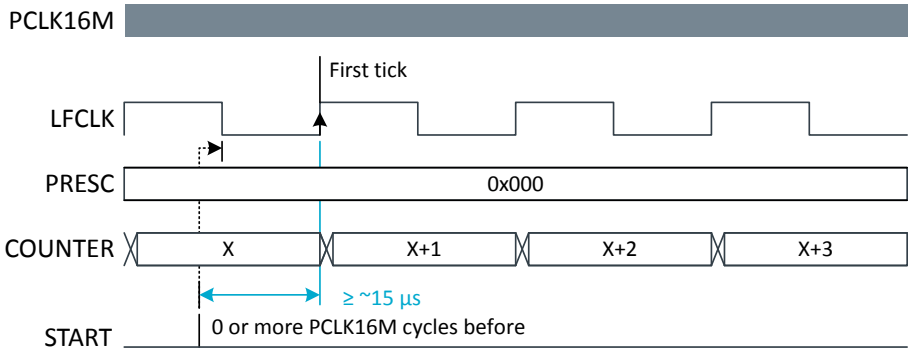

Figure 187: Timing diagram - JITTER_START-

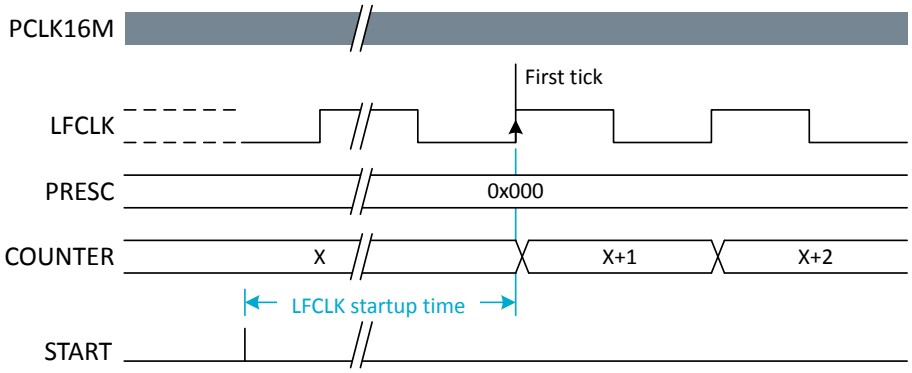

Figure 188: Timing diagram - JITTER_START+

The following tables summarize jitter introduced for tasks and events. Any 32.768 kHz clock jitter will come in addition to these numbers.

Task	Delay
CLEAR, START, STOP, TRIGOVRFLOW	+15 to 46 µs

Table 135: RTC jitter magnitudes on tasks

Operation/Function	Jitter
START to COUNTER increment	± 15 µs
COMPARE to COMPARE [16]	± 62.5 ns

Table 136: RTC jitter magnitudes on events

7.28.10 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50014000 0x40014000	APPLICATION	RTC	RTC0 : S RTC0 : NS	US	NA	Real time counter 0	
0x50015000 0x40015000	APPLICATION	RTC	RTC1 : S RTC1 : NS	US	NA	Real time counter 1	
0x41011000	NETWORK	RTC	RTC0	NS	NA	Real-time counter 0	
0x41016000	NETWORK	RTC	RTC1	NS	NA	Real-time counter 1	

Table 137: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start RTC counter
TASKS_STOP	0x004		Stop RTC counter
TASKS_CLEAR	0x008		Clear RTC counter
TASKS_TRIGOVRFLW	0x00C		Set counter to 0xFFFFF0
TASKS_CAPTURE[n]	0x040		Capture RTC counter to CC[n] register
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
SUBSCRIBE_CLEAR	0x088		Subscribe configuration for task CLEAR
SUBSCRIBE_TRIGOVRFLW	0x08C		Subscribe configuration for task TRIGOVRFLW
SUBSCRIBE_CAPTURE[n]	0x0C0		Subscribe configuration for task CAPTURE[n]
EVENTS_TICK	0x100		Event on counter increment
EVENTS_OVRFLW	0x104		Event on counter overflow
EVENTS_COMPARE[n]	0x140		Compare event on CC[n] match
PUBLISH_TICK	0x180		Publish configuration for event TICK
PUBLISH_OVRFLW	0x184		Publish configuration for event OVRFLW
PUBLISH_COMPARE[n]	0x1C0		Publish configuration for event COMPARE[n]
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
EVTEN	0x340		Enable or disable event routing
EVTENSET	0x344		Enable event routing
EVTENCLR	0x348		Disable event routing
COUNTER	0x504		Current counter value
PRESCALER	0x508		12-bit prescaler for counter frequency (32768 / (PRESCALER + 1)). Must be written when RTC is stopped.

[16] Assumes RTC runs continuously between these events.

Register	Offset	Security	Description
CC[n]	0x540		Compare register n

Table 138: Register overview

7.28.10.1 TASKS_START

Address offset: 0x000

Start RTC counter

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start RTC counter
			Trigger	1	Trigger task

7.28.10.2 TASKS_STOP

Address offset: 0x004

Stop RTC counter

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop RTC counter
			Trigger	1	Trigger task

7.28.10.3 TASKS_CLEAR

Address offset: 0x008

Clear RTC counter

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_CLEAR			Clear RTC counter
			Trigger	1	Trigger task

7.28.10.4 TASKS_TRIGOVRFLW

Address offset: 0x00C

Set counter to 0xFFFFF0

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_TRIGOVRFLW			Set counter to 0xFFFFF0
			Trigger	1	Trigger task

7.28.10.5 TASKS_CAPTURE[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Capture RTC counter to CC[n] register

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_CAPTURE			Capture RTC counter to CC[n] register
			Trigger	1	Trigger task

7.28.10.6 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.28.10.7 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.28.10.8 SUBSCRIBE_CLEAR

Address offset: 0x088

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CLEAR will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.28.10.9 SUBSCRIBE_TRIGOVRFLW

Address offset: 0x08C

Subscribe configuration for task TRIGOVRFLW

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task TRIGOVRFLW will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.28.10.10 SUBSCRIBE_CAPTURE[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task CAPTURE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CAPTURE[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.28.10.11 EVENTS_TICK

Address offset: 0x100

Event on counter increment

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TICK			Event on counter increment
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.28.10.12 EVENTS_OVRFLW

Address offset: 0x104

Event on counter overflow

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_OVRFLW			Event on counter overflow
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.28.10.13 EVENTS_COMPARE[n] (n=0..3)

Address offset: 0x140 + (n × 0x4)

Compare event on CC[n] match

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_COMPARE			Compare event on CC[n] match
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.28.10.14 PUBLISH_TICK

Address offset: 0x180

Publish configuration for event TICK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TICK will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.28.10.15 PUBLISH_OVRFLW

Address offset: 0x184

Publish configuration for event OVRFLW

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event OVRFLW will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.28.10.16 PUBLISH_COMPARE[n] (n=0..3)

Address offset: 0x1C0 + (n × 0x4)

Publish configuration for event COMPARE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event COMPARE[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.28.10.17 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A-D	RW	COMPARE[i]_CLEAR (i=0..3)			Shortcut between event COMPARE[i] and task CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.28.10.18 INTENSET

Address offset: 0x304

Enable interrupt

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

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

7.28.10.19 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to disable interrupt for event TICK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	OVRFLW			Write '1' to disable interrupt for event OVRFLW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C-F	RW	COMPARE[i] (i=0..3)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.28.10.20 EVTEN

Address offset: 0x340

Enable or disable event routing

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TICK			Enable or disable event routing for event TICK
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	OVRFLW			Enable or disable event routing for event OVRFLW
			Disabled	0	Disable
			Enabled	1	Enable
C-F	RW	COMPARE[i] (i=0..3)			Enable or disable event routing for event COMPARE[i]
			Disabled	0	Disable
			Enabled	1	Enable

7.28.10.21 EVTENSET

Address offset: 0x344

Enable event routing

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to enable event routing for event TICK
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Set	1	Enable
B	RW	OVRFLW			Write '1' to enable event routing for event OVRFLW
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Set	1	Enable
C-F	RW	COMPARE[i] (i=0..3)			Write '1' to enable event routing for event COMPARE[i]
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Set	1	Enable

7.28.10.22 EVTENCLR

Address offset: 0x348

Disable event routing

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	TICK			Write '1' to disable event routing for event TICK
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Clear	1	Disable
B	RW	OVRFLW			Write '1' to disable event routing for event OVRFLW
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Clear	1	Disable
C-F	RW	COMPARE[i] (i=0..3)			Write '1' to disable event routing for event COMPARE[i]
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
			Clear	1	Disable

7.28.10.23 COUNTER

Address offset: 0x504

Current counter value

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

ID	R/W	Field	Value ID	Value	Description
A	R	COUNTER			Counter value

7.28.10.24 PRESCALER

Address offset: 0x508

12-bit prescaler for counter frequency (32768 / (PRESCALER + 1)). Must be written when RTC is stopped.

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

7.28.10.25 CC[n] (n=0..3)

Address offset: 0x540 + (n × 0x4)

Compare register n

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

7.28.11 Electrical specification

7.29 SAADC — Successive approximation analog-to-digital converter

SAADC is a differential successive approximation register (SAR) analog-to-digital converter (ADC).

The main features of SAADC are the following:

- 8/10/12-bit resolution, 14-bit resolution with oversampling
- Multiple analog inputs:
  - **AIN0** to **AIN7** pins
  - **VDD** pin
  - VDDHDIV5 (through **VDDH** pin)
- 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
  - Each channel can be configured to select any of the above analog inputs
- 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 a low-power 32.768 kHz RTC or more accurate 1/16 MHz timers

- One-shot conversion mode to sample a single channel
- Scan mode to sample a series of channels in sequence with configurable sample delay
- Support for direct sample transfer to RAM using EasyDMA
- Interrupts on single sample and full buffer events
- Samples stored as 16-bit two's complement values for differential and single-ended sampling
- Continuous sampling without the need of an external timer
- Internal resistor string
- On-the-fly limit checking

7.29.1 Shared resources

The ADC can coexist with COMP 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.

7.29.2 Overview

The ADC supports up to eight external analog input channels. It can be operated in One-shot mode with sampling under software control, or Continuous 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 one of the following:

- **AIN0** to **AIN7** pins
- **VDD** pin
- **VDDHDIV5** (through VDDH pin)

Channels can be sampled individually in One-shot or Continuous sampling modes. Using Scan mode, multiple channels can be sampled in sequence. Channels can also be oversampled to improve noise performance.

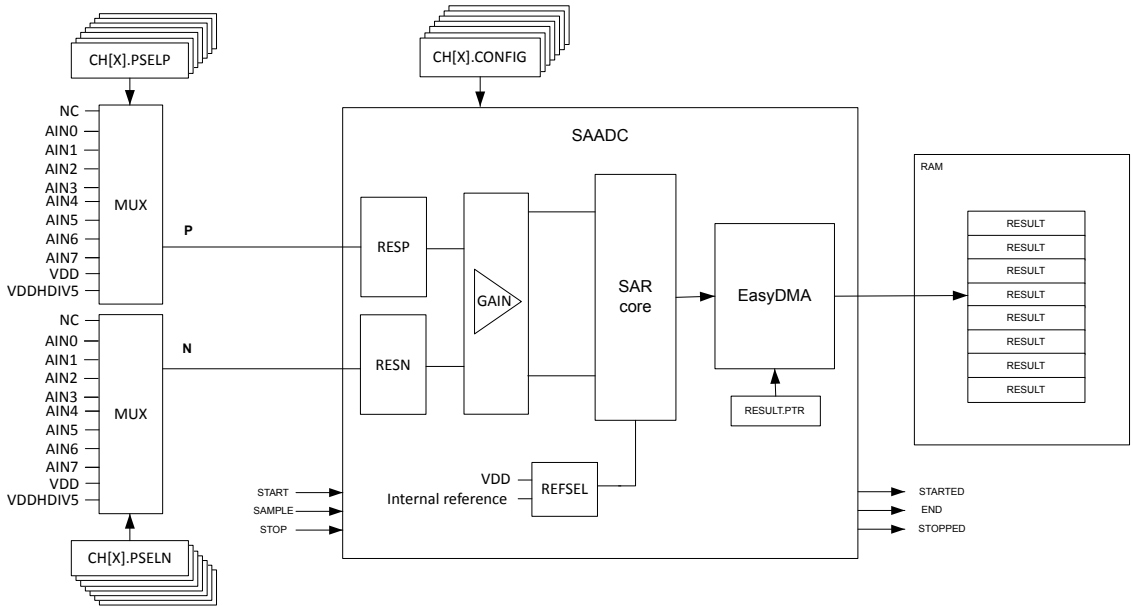

Figure 189: Simplified ADC block diagram

Internally, the ADC is always a differential analog-to-digital converter, but by default it is configured with single-ended input `SE` in the MODE field of the CH[n].CONFIG register. In Single-ended (SE) mode, the negative input will be shorted to ground internally.

In Single-ended mode, the assumption 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.

This can be reduced by configuring SAADC to use differential measurent setting the MODE field of the CH[n].CONFIG register to `Diff` (differential).

7.29.3 Digital output

The output result of the ADC depends on the settings in the CH[n].CONFIG and RESOLUTION registers as shown in the following equation.

$$RESULT = [V(P) - V(N)] * GAIN/REFERENCE * 2^{(RESOLUTION - m)}$$

Variable	Description
V(P)	Voltage at input P
V(N)	Voltage at input N
GAIN	Selected gain setting
m	Mode setting (Use m=0 if CONFIG.MODE=SE, or m=1 if CONFIG.MODE=Diff)
REFERENCE	Selected reference voltage

Table 139: Equation variables

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 ±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, it is recommended to run CALIBRATEOFFSET at regular intervals. The CALIBRATEDONE, DONE, and RESULTDONE events will be generated when the calibration has completed.

7.29.4 Analog inputs and channels

Up to eight analog input channels, CH[n](n=0..7), can be configured.

See Shared resources on page 520 for shared input with comparators.

Any 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 the 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 the CH[n].CONFIG register.

Note: Channels selected for COMP 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.

7.29.5 Operation modes

The ADC input configuration supports One-shot mode, Continuous mode, and Scan mode.

The ADC indicates a single ongoing conversion via the register STATUS on page 540. During Scan mode, oversampling, or Continuous modes, more than a single conversion take place in the ADC. As a consequence, the value reflected in the STATUS register will toggle at the end of each single conversion.

Note: Scan mode and oversampling cannot be combined.

7.29.5.1 One-shot mode

One-shot operation is configured by enabling only one of the available channels defined by the CH[n].PSELP, CH[n].PSELN, and CH[n].CONFIG registers.

Once a SAMPLE task is triggered, 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. Both events may occur before the actual value has been transferred into RAM by EasyDMA. For more information, see EasyDMA on page 523.

7.29.5.2 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 system.

Care shall be taken to ensure that the sample rate fulfills the following criteria, depending on how many channels are active.

$$f_{SAMPLE} < 1/(t_{ACQ} + t_{conv})$$

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. Both events may occur before the actual value has been transferred into RAM by EasyDMA.

7.29.5.3 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 mode 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 $2^{OVERSAMPLE}$ number of times before the result is written to RAM. This can be achieved by the following:

- Configuring a fixed sampling rate using the local timer or a general purpose timer and the PPI system to trigger a SAMPLE task
- Triggering SAMPLE $2^{OVERSAMPLE}$ times from software
- Enabling BURST mode

CH[n].CONFIG.BURST can be enabled to avoid setting SAMPLE task $2^{OVERSAMPLE}$ times. With BURST = 1 the ADC will sample the input $2^{OVERSAMPLE}$ times as fast as it can (actual timing: $<(t_{ACQ}+t_{CONV})\times 2^{OVERSAMPLE}$). 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.

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. Both events may occur before the actual value has been transferred into RAM by EasyDMA.

7.29.5.4 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 given by the following equation:

```
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. Both events may occur before the actual values have been transferred into RAM by EasyDMA.

The following figure shows an example of the placement of results 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] 1st result	CH[1] 1st result
RESULT.PTR + 4	CH[1] 2nd result	CH[5] 1st result
RESULT.PTR + 8	CH[5] 2nd result	CH[2] 2nd result
	(…)	
RESULT.PTR + 2*(RESULT.MAXCNT – 2)	CH[5] last result	CH[2] last result

Figure 190: Example of RAM placement (even RESULT.MAXCNT), channels 1, 2 and, 5 enabled

The following figure shows an example of the placement of results 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] 1st result	CH[1] 1st result
RESULT.PTR + 4	CH[1] 2nd result	CH[5] 1st result
RESULT.PTR + 8	CH[5] 2nd result	CH[2] 2nd result
	(…)	
RESULT.PTR + 2*(RESULT.MAXCNT – 1)		CH[5] last result

Figure 191: Example of RAM placement (odd RESULT.MAXCNT), channels 1, 2 and, 5 enabled

7.29.6 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 ADC on page 524. 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.

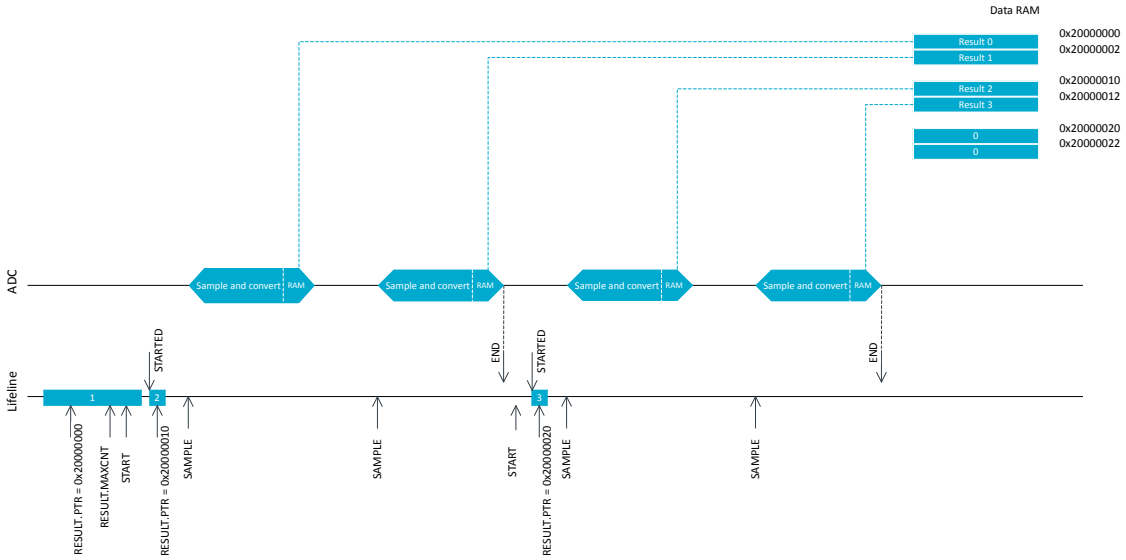

Figure 192: ADC

If the RESULT.PTR is not pointing to a RAM region accessible from the peripheral, an EasyDMA transfer may result in a HardFault and/or memory corruption. See Memory on page 18 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, SAMPLE tasks can be triggered once the START task is triggered. The END event is generated when the number of samples transferred to memory reaches the value specified by RESULT.MAXCNT. After an END event, the START task needs to be triggered again before new samples can be taken. By specifying RESULT.MAXCNT ≥ number of channels enabled, the size of the Result buffer should be large enough for a minimum of one result from each of the enabled channels. See Scan mode on page 523 for more information.

7.29.7 Resistor ladder

The ADC has an internal resistor string for positive and negative input.

The resistors in the following figure are controlled in the CH[n].CONFIG.RESP and CH[n].CONFIG.RESN registers.

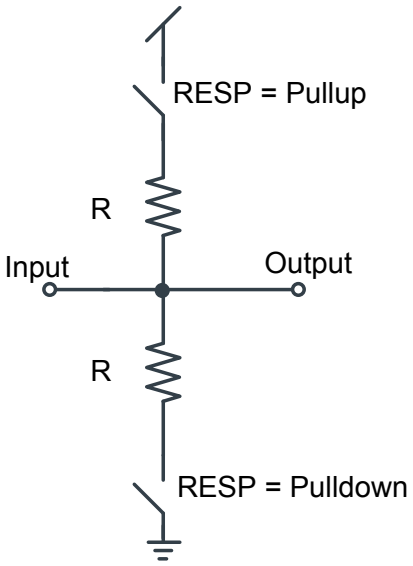

Figure 193: Resistor ladder for positive input

7.29.8 Reference

The ADC can use the following two references, controlled in the REFSEL field of the CH[n].CONFIG register.

- Internal reference
- VDD as reference

The internal reference results in an input range of ±0.6 V on the ADC core. VDD as reference results in an input range of ±VDD/4 on the ADC core. The gain block can be used to change the effective input range of the ADC.

```
Input range = (+- 0.6 V or +-VDD/4)/Gain
```

For example, choosing VDD as reference, single ended input (grounded negative input), and a gain of 1/4 the input range is the following:

```
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 is the following:

```
Input range = (0.6 V)/(1/6) = 3.6 V
```

The **AIN0** - **AIN7** inputs cannot exceed VDD, or be lower than VSS.

7.29.9 Acquisition time

To sample the input voltage, the ADC connects a capacitor to the input.

The acquisition time indicates how long the capacitor is connected, see TACQ field in register CH[n].CONFIG (n=0..7) on page 541. The required acquisition time depends on the source (R_{source}) resistance. For high source resistance, the acquisition time should be increased. See the following table for more information.

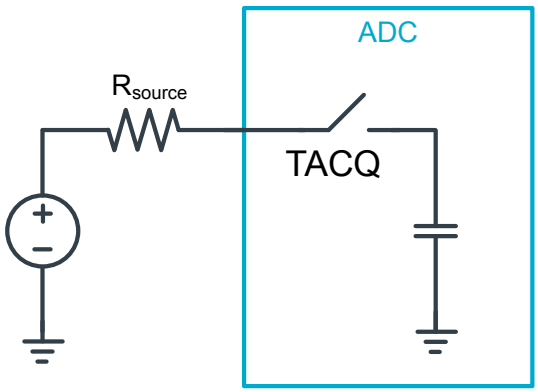

Figure 194: Simplified ADC sample network

TACQ [µs]	Maximum source resistance [kOhm]
3	10
5	40
10	100
15	200
20	400
40	800

Table 140: Acquisition time

When using **VDDHDIV5** as input, the acquisition time needs to be 10 µs or higher.

7.29.10 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.

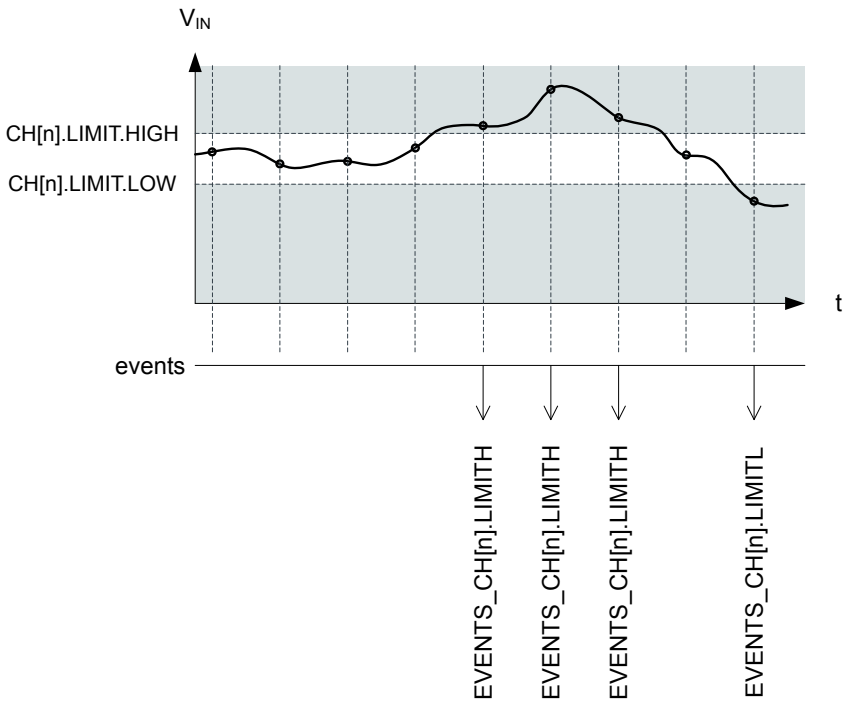

Figure 195: Example of limits monitoring on channel 'n'

The CH[n].LIMIT.HIGH shall always be higher than or equal to CH[n].LIMIT.LOW . In other words, an event can be generated only when the input signal has been sampled outside of the defined limits. It is not possible to generate an event when the input signal is inside a defined range by swapping high and low limits.

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.

7.29.11 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5000E000 0x4000E000	APPLICATION	SAADC	SAADC : S SAADC : NS	US	SA	Successive approximation analog-to-digital converter	

Table 141: Instances

Register	Offset	Security	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 ongoing conversion
TASKS_CALIBRATEOFFSET	0x00C		Starts offset auto-calibration
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_SAMPLE	0x084		Subscribe configuration for task SAMPLE
SUBSCRIBE_STOP	0x088		Subscribe configuration for task STOP
SUBSCRIBE_CALIBRATEOFFSET	0x08C		Subscribe configuration for task CALIBRATEOFFSET
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_CALIBRATEDONE	0x110		Calibration is complete
EVENTS_STOPPED	0x114		The ADC has stopped
EVENTS_CH[n].LIMITH	0x118		Last results is equal or above CH[n].LIMIT.HIGH
EVENTS_CH[n].LIMITL	0x11C		Last results is equal or below CH[n].LIMIT.LOW
PUBLISH_STARTED	0x180		Publish configuration for event STARTED
PUBLISH_END	0x184		Publish configuration for event END
PUBLISH_DONE	0x188		Publish configuration for event DONE
PUBLISH_RESULTDONE	0x18C		Publish configuration for event RESULTDONE
PUBLISH_CALIBRATEDONE	0x190		Publish configuration for event CALIBRATEDONE
PUBLISH_STOPPED	0x194		Publish configuration for event STOPPED
PUBLISH_CH[n].LIMITH	0x198		Publish configuration for event CH[n].LIMITH
PUBLISH_CH[n].LIMITL	0x19C		Publish configuration for event CH[n].LIMITL
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
STATUS	0x400		Status
ENABLE	0x500		Enable or disable ADC
CH[n].PSELP	0x510		Input positive pin selection for CH[n]
CH[n].PSELN	0x514		Input negative pin selection for CH[n]
CH[n].CONFIG	0x518		Input configuration for CH[n]
CH[n].LIMIT	0x51C		High/low limits for event monitoring a channel
RESOLUTION	0x5F0		Resolution configuration

Register	Offset	Security	Description
OVERSAMPLE	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.
SAMPLERATE	0x5F8		Controls normal or continuous sample rate
RESULT.PTR	0x62C		Data pointer
RESULT.MAXCNT	0x630		Maximum number of buffer words to transfer
RESULT.AMOUNT	0x634		Number of buffer words transferred since last START

Table 142: Register overview

7.29.11.1 TASKS_START

Address offset: 0x000

Start the ADC and prepare the result buffer in RAM

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start the ADC and prepare the result buffer in RAM
			Trigger	1	Trigger task

7.29.11.2 TASKS_SAMPLE

Address offset: 0x004

Take one ADC sample, if scan is enabled all channels are sampled

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_SAMPLE			Take one ADC sample, if scan is enabled all channels are sampled
			Trigger	1	Trigger task

7.29.11.3 TASKS_STOP

Address offset: 0x008

Stop the ADC and terminate any ongoing conversion

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop the ADC and terminate any ongoing conversion
			Trigger	1	Trigger task

7.29.11.4 TASKS_CALIBRATEOFFSET

Address offset: 0x00C

Starts offset auto-calibration

Do not trigger when the ADC has been started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_CALIBRATEOFFSET			Starts offset auto-calibration
					Do not trigger when the ADC has been started
			Trigger	1	Trigger task

7.29.11.5 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.29.11.6 SUBSCRIBE_SAMPLE

Address offset: 0x084

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SAMPLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.29.11.7 SUBSCRIBE_STOP

Address offset: 0x088

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.29.11.8 SUBSCRIBE_CALIBRATEOFFSET

Address offset: 0x08C

Subscribe configuration for task CALIBRATEOFFSET

Do not trigger when the ADC has been started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CALIBRATEOFFSET will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.29.11.9 EVENTS_STARTED

Address offset: 0x100

The ADC has started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			The ADC has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.10 EVENTS_END

Address offset: 0x104

The ADC has filled up the Result 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			The ADC has filled up the Result buffer
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.11 EVENTS_DONE

Address offset: 0x108

A conversion task has been completed. Depending on the mode, multiple conversions might be needed for a result to be transferred to RAM.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_DONE			A conversion task has been completed. Depending on the mode, multiple conversions might be needed for a result to be transferred to RAM.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.12 EVENTS_RESULTDONE

Address offset: 0x10C

A result is ready to get transferred to RAM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RESULTDONE			A result is ready to get transferred to RAM
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.13 EVENTS_CALIBRATEDONE

Address offset: 0x110

Calibration is complete

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CALIBRATEDONE			Calibration is complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.14 EVENTS_STOPPED

Address offset: 0x114

The ADC has stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			The ADC has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.15 EVENTS_CH[n].LIMITH (n=0..7)

Address offset: 0x118 + (n × 0x8)

Last results is equal or above CH[n].LIMIT.HIGH

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	LIMITH			Last results is equal or above CH[n].LIMIT.HIGH
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.16 EVENTS_CH[n].LIMITL (n=0..7)

Address offset: 0x11C + (n × 0x8)

Last results is equal or below CH[n].LIMIT.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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	LIMITL			Last results is equal or below CH[n].LIMIT.LOW
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.29.11.17 PUBLISH_STARTED

Address offset: 0x180

Publish configuration for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.18 PUBLISH_END

Address offset: 0x184

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.19 PUBLISH_DONE

Address offset: 0x188

Publish configuration for event DONE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.20 PUBLISH_RESULTDONE

Address offset: 0x18C

Publish configuration for event RESULTDONE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RESULTDONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.21 PUBLISH_CALIBRATEDONE

Address offset: 0x190

Publish configuration for event CALIBRATEDONE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CALIBRATEDONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.22 PUBLISH_STOPPED

Address offset: 0x194

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.23 PUBLISH_CH[n].LIMITH (n=0..7)

Address offset: 0x198 + (n × 0x8)

Publish configuration for event CH[n].LIMITH

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CH[n].LIMITH will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.24 PUBLISH_CH[n].LIMITL (n=0..7)

Address offset: 0x19C + (n × 0x8)

Publish configuration for event CH[n].LIMITL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CH[n].LIMITL will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.29.11.25 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Enable or disable interrupt for event STARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	END			Enable or disable interrupt for event END
			Disabled	0	Disable
			Enabled	1	Enable

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

ID	R/W	Field	Value ID	Value	Description
C	RW	DONE			Enable or disable interrupt for event DONE
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	RESULTDONE			Enable or disable interrupt for event RESULTDONE
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	CALIBRATEDONE			Enable or disable interrupt for event CALIBRATEDONE
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	CH0LIMITH			Enable or disable interrupt for event CH0LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	CH0LIMITL			Enable or disable interrupt for event CH0LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	CH1LIMITH			Enable or disable interrupt for event CH1LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	CH1LIMITL			Enable or disable interrupt for event CH1LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	CH2LIMITH			Enable or disable interrupt for event CH2LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	CH2LIMITL			Enable or disable interrupt for event CH2LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	CH3LIMITH			Enable or disable interrupt for event CH3LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	CH3LIMITL			Enable or disable interrupt for event CH3LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	CH4LIMITH			Enable or disable interrupt for event CH4LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	CH4LIMITL			Enable or disable interrupt for event CH4LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
Q	RW	CH5LIMITH			Enable or disable interrupt for event CH5LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	CH5LIMITL			Enable or disable interrupt for event CH5LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	CH6LIMITH			Enable or disable interrupt for event CH6LIMITH
			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				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Enable
T	RW	CH6LIMITL			Enable or disable interrupt for event CH6LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
U	RW	CH7LIMITH			Enable or disable interrupt for event CH7LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
V	RW	CH7LIMITL			Enable or disable interrupt for event CH7LIMITL
			Disabled	0	Disable
			Enabled	1	Enable

7.29.11.26 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to enable interrupt for event DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to enable interrupt for event RESULTDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to enable interrupt for event CALIBRATEDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to enable interrupt for event CH0LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to enable interrupt for event CH0LIMITL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to enable interrupt for event CH1LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to enable interrupt for event CH1LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to enable interrupt for event CH2LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to enable interrupt for event CH2LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to enable interrupt for event CH3LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to enable interrupt for event CH3LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	CH4LIMITH			Write '1' to enable interrupt for event CH4LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to enable interrupt for event CH4LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to enable interrupt for event CH5LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to enable interrupt for event CH5LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to enable interrupt for event CH6LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to enable interrupt for event CH6LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
U	RW	CH7LIMITH			Write '1' to enable interrupt for event CH7LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to enable interrupt for event CH7LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.29.11.27 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to disable interrupt for event DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to disable interrupt for event RESULTDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to disable interrupt for event CALIBRATEDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to disable interrupt for event CH0LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to disable interrupt for event CH0LIMITL
			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				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to disable interrupt for event CH1LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to disable interrupt for event CH1LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to disable interrupt for event CH2LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to disable interrupt for event CH2LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to disable interrupt for event CH3LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to disable interrupt for event CH3LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	CH4LIMITH			Write '1' to disable interrupt for event CH4LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to disable interrupt for event CH4LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to disable interrupt for event CH5LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to disable interrupt for event CH5LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to disable interrupt for event CH6LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to disable interrupt for event CH6LIMITL
			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				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
U	RW	CH7LIMITH			Write '1' to disable interrupt for event CH7LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to disable interrupt for event CH7LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.29.11.28 STATUS

Address offset: 0x400

Status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	STATUS			Status
			Ready	0	ADC is ready. No ongoing conversion.
			Busy	1	ADC is busy. Single conversion in progress.

7.29.11.29 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	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].PSELP and CH[n].PSELN registers.

7.29.11.30 CH[n].PSELP (n=0..7)

Address offset: 0x510 + (n × 0x10)

Input positive pin selection for CH[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PSELP			Analog positive input channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD
			VDDHDIV5	0xD	VDDH/5

7.29.11.31 CH[n].PSELN (n=0..7)

Address offset: 0x514 + (n × 0x10)

Input negative pin selection for CH[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PSELN			Analog negative input, enables differential channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD
			VDDHDIV5	0xD	VDDH/5

7.29.11.32 CH[n].CONFIG (n=0..7)

Address offset: 0x518 + (n × 0x10)

Input configuration for CH[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E E E D C C C B B A A
Reset 0x00020000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	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

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					G F E E E D C C C B B A A
Reset 0x00020000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
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 µs
			5us	1	5 µs
			10us	2	10 µs
			15us	3	15 µs
			20us	4	20 µs
			40us	5	40 µs
F	RW	MODE			Enable differential mode
			SE	0	Single-ended, PSELN will be ignored, negative input to 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.

7.29.11.33 CH[n].LIMIT (n=0..7)

Address offset: 0x51C + (n × 0x10)

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
Reset 0x7FFF8000					**0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LOW		[-32768 to +32767]	Low level limit
B	RW	HIGH		[-32768 to +32767]	High level limit

7.29.11.34 RESOLUTION

Address offset: 0x5F0

Resolution configuration

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

ID	R/W	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

7.29.11.35 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.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	OVERSAMPLE			Oversample control
			Bypass	0	Bypass oversampling
			Over2x	1	Oversample 2x
			Over4x	2	Oversample 4x
			Over8x	3	Oversample 8x
			Over16x	4	Oversample 16x
			Over32x	5	Oversample 32x
			Over64x	6	Oversample 64x
			Over128x	7	Oversample 128x
			Over256x	8	Oversample 256x

7.29.11.36 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	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)

7.29.11.37 RESULT.PTR

Address offset: 0x62C

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.29.11.38 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	MAXCNT			Maximum number of buffer words to transfer

7.29.11.39 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	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.

7.29.12 Electrical specification

7.29.12.1 SAADC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
DNL_{10}	Differential non-linearity, 10-bit resolution				LSB10b
INL_{10}	Integral non-linearity, 10-bit resolution				LSB10b
V_{OS}	Differential offset error (calibrated), 10-bit resolution [17]				LSB10b
$E_{VDDHDIV5}$	Error on VDDHDIV5 input		±1		%
C_{EG}	Gain error temperature coefficient	-0.05	0.02	0.05	%/°C
f_{SAMPLE}	Maximum sampling rate			200	kHz
$t_{ACQ,10k}$	Acquisition time (configurable), source Resistance ≤ 10 kΩ		3		µs
$t_{ACQ,40k}$	Acquisition time (configurable), source Resistance ≤ 40 kΩ		5		µs
$t_{ACQ,100k}$	Acquisition time (configurable), source Resistance ≤ 100 kΩ		10		µs
$t_{ACQ,200k}$	Acquisition time (configurable), source Resistance ≤ 200 kΩ		15		µs
$t_{ACQ,400k}$	Acquisition time (configurable), source Resistance ≤ 400 kΩ		20		µs
$t_{ACQ,800k}$	Acquisition time (configurable), source Resistance ≤ 800 kΩ		40		µs
t_{CONV}	Conversion time		2		µs

[17] Digital output code at zero volt differential input.

Symbol	Description	Min.	Typ.	Max.	Units
$E_{G1/6}$	Error[18] for Gain = 1/6				%
$E_{G1/4}$	Error[18] for Gain = 1/4				%
$E_{G1/2}$	Error[18] for Gain = 1/2				%
E_{G1}	Error[18] for Gain = 1				%
C_{SAMPLE}	Sample and hold capacitance at maximum gain[19]		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, HFXO, 200 ksps				Bit
S_{NDR}	Peak signal to noise and distortion ratio, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, HFXO, 200 ksps				dB
S_{FDR}	Spurious free dynamic range, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, HFXO, 200 ksps				dBc
R_{LADDER}	Ladder resistance		160		kΩ

7.29.13 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. The best ADC performance is achieved using START timing based on the TIMER module, HFXO clock source, and Constant Latency mode.

7.30 SPIM — Serial peripheral interface master with EasyDMA

The SPI master can communicate with multiple SPI slaves using individual chip select signals for each slave.

The the main features of SPIM are:

- EasyDMA direct transfer to/from RAM
- SPI mode 0-3
- Individual selection of I/O pins
- Optional D/CX output line for distinguishing between command and data bytes

[18] Temperature drift is not included.
[19] Maximum gain corresponds to highest capacitance.

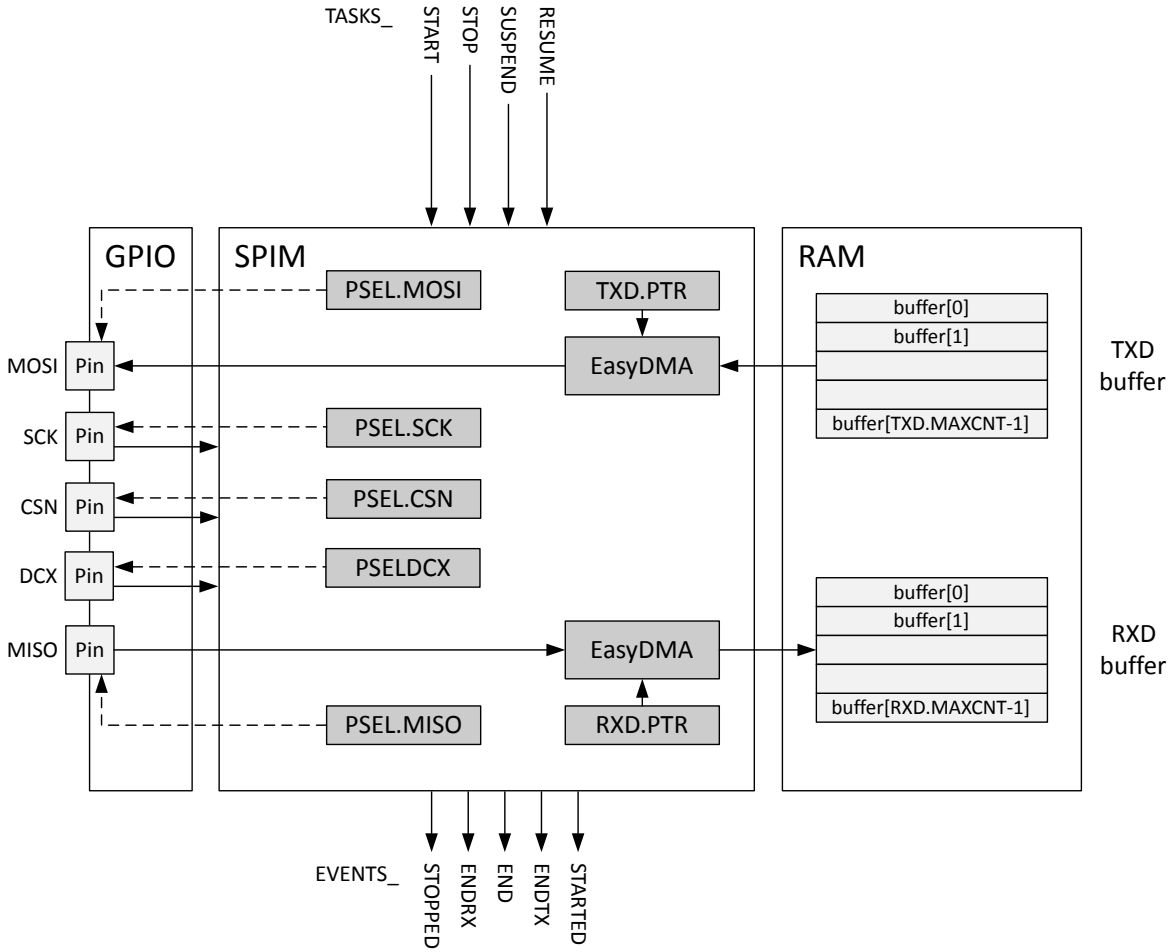

Figure 196: SPIM — SPI master with EasyDMA

7.30.1 SPI master transaction sequence

An SPI master transaction is started by triggering the START task. When started, a number of bytes will be transmitted/received on MOSI/MISO.

The following figure illustrates an SPI master transaction.

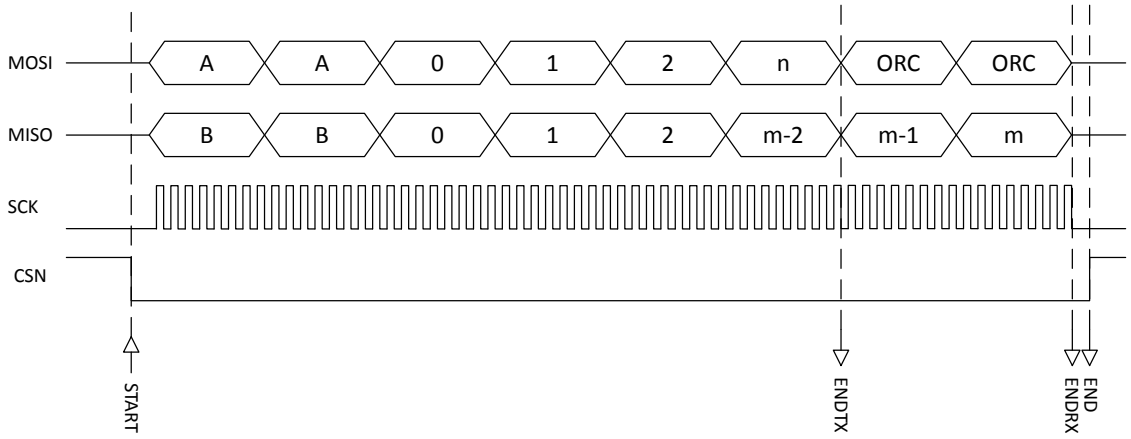

Figure 197: SPI master transaction

The ENDTX is generated when all bytes in buffer TXD.PTR on page 561 are transmitted. The number of bytes in the transmit buffer is specified in register TXD.MAXCNT on page 561. The ENDRX event will be generated when buffer RXD.PTR on page 560 is full, that is when the number of bytes specified in register RXD.MAXCNT on page 560 have been received. The transaction stops automatically after all bytes have been transmitted/received. When the maximum number of bytes in receive buffer is larger than the number of bytes in the transmit buffer, the contents of register ORC on page 564 will be transmitted after the last byte in the transmit buffer has been transmitted.

The END event will be generated after both the ENDRX and ENDTX events have been generated.

The SPI master can be stopped by triggering the STOP task. A STOPPED event is generated when the SPI master has stopped. If the STOP task is triggered in the middle of a transaction, SPIM will complete the transmission/reception of the current byte before stopping. The STOPPED event is generated even if the STOP task is triggered while there is no ongoing transaction.

If the ENDTX event has not already been generated when the SPI master has come to a stop, the ENDTX event will be generated even if all bytes in the buffer TXD.PTR on page 561 have not been transmitted.

If the ENDRX event has not already been generated when the SPI master has come to a stop, the ENDRX event will be generated even if the buffer RXD.PTR on page 560 is not full.

A transaction can be suspended and resumed using the SUSPEND and RESUME tasks, receptively. When the SUSPEND task is triggered, the SPI master will complete transmitting and receiving the current ongoing byte before it is suspended.

7.30.2 D/CX functionality

Some SPI slaves, for example display drivers, require an additional signal from the SPI master to distinguish between command and data bytes. For display drivers this line is often called D/CX.

The SPIM provides support for such a D/CX output line. The D/CX line is set low during transmission of command bytes and high during transmission of data bytes.

The D/CX pin number is selected using PSELDCX on page 563 and the number of command bytes preceding the data bytes is configured using DCXCNT on page 564.

It is not allowed to write to the DCXCNT on page 564 during an ongoing transmission.

The following figure shows the use of D/CX, using SPIM.DCXCNT=1.

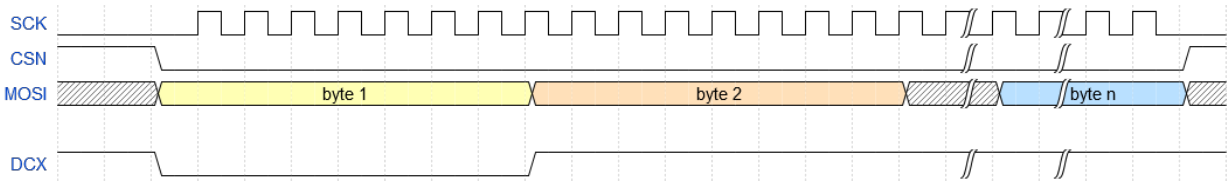

Figure 198: D/CX example. SPIM.DCXCNT = 1.

7.30.3 Pin configuration

The SCK, CSN, DCX, MOSI, and MISO signals associated with the SPIM are mapped to physical pins according to the configuration specified in the PSEL.n registers.

The contents of registers PSEL.SCK on page 558, PSEL.CSN on page 559, PSELDCX on page 563, PSEL.MOSI on page 559, and PSEL.MISO on page 559 are only used when the SPIM is enabled and retained only as long as the device is in System ON mode. The PSEL.n registers can only be configured when the SPIM is disabled. Enabling/disabling is done using register ENABLE on page 558.

To ensure correct behavior, the pins used by the SPIM must be configured in the GPIO peripheral as described in GPIO configuration on page 548 before the SPIM is enabled. Make sure to activate the dedicated peripheral setting of the GPIO pin. See GPIO — General purpose input/output on page 219 for details on how to assign pins between cores, peripherals, or subsystems. For pin recommendations, see Pin assignments on page 783.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI master signal	SPI master pin	Direction	Output value
SCK	As specified in PSEL.SCK on page 558	Output	Same as CONFIG.CPOL
CSN	As specified in PSEL.CSN on page 559	Output	Same as CONFIG.CPOL
DCX	As specified in PSELDCX on page 563	Output	1
MOSI	As specified in PSEL.MOSI on page 559	Output	0
MISO	As specified in PSEL.MISO on page 559	Input	Not applicable

Table 143: GPIO configuration

Some SPIM instances do not support automatic control of CSN, and for those the available GPIO pins need to be used to control CSN directly. See Instances on page 550 for information about what features are supported in the various SPIM instances.

The SPIM supports SPI modes 0 through 3. The clock polarity (CPOL) and the clock phase (CPHA) are configured in register CONFIG on page 562.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Active High)	0 (Leading)
SPI_MODE1	0 (Active High)	1 (Trailing)
SPI_MODE2	1 (Active Low)	0 (Leading)
SPI_MODE3	1 (Active Low)	1 (Trailing)

Table 144: SPI modes

7.30.4 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 Peripherals on page 145 for details on peripherals and their IDs.

7.30.5 EasyDMA

SPIM implements EasyDMA for accessing RAM without CPU involvement.

SPIM implements the following EasyDMA channels.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 145: SPIM EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 149.

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 SPI master will automatically stop transmitting after TXD.MAXCNT bytes have been transmitted and RXD.MAXCNT bytes have been received. If RXD.MAXCNT is larger than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register. If TXD.MAXCNT is larger than RXD.MAXCNT, the additional received bytes will be discarded.

The ENDRX/ENDTX events 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.

If several AHB bus masters try to access the same AHB slave at the same time, AHB bus congestion might occur, and the behavior of the EasyDMA channel will depend on the SPIM instance. Refer to Instances on page 550 for information about what behavior is supported in the various instances.

7.30.6 Low power

To ensure lowest possible power consumption when the peripheral is not needed stop and disable SPIM.

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.

7.30.7 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	APPLICATION	SPIM	SPIM0 : S SPIM0 : NS	US	SA	SPI master 0	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention
0x50009000 0x40009000	APPLICATION	SPIM	SPIM1 : S SPIM1 : NS	US	SA	SPI master 1	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention
0x5000A000 0x4000A000	APPLICATION	SPIM	SPIM4 : S SPIM4 : NS	US	SA	SPI master 4 (high-speed)	Up to 32 Mbps SPI when using dedicated pins
0x5000B000 0x4000B000	APPLICATION	SPIM	SPIM2 : S SPIM2 : NS	US	SA	SPI master 2	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention
0x5000C000 0x4000C000	APPLICATION	SPIM	SPIM3 : S SPIM3 : NS	US	SA	SPI master 3	Not supported: > 8 Mbps data rate, CSNPOL register, DCX functionality, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention
0x41013000	NETWORK	SPIM	SPIM0	NS	NA	SPI master 0	Not supported: > 8 Mbps data rate, IFTIMING.x registers, hardware CSN control (PSEL.CSN), stalling mechanism during AHB bus contention

Table 146: Instances

Register	Offset	Security	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
SUBSCRIBE_START	0x090		Subscribe configuration for task START
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP

Register	Offset	Security	Description
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
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
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_END	0x198		Publish configuration for event END
PUBLISH_ENDTX	0x1A0		Publish configuration for event ENDTX
PUBLISH_STARTED	0x1CC		Publish configuration for event STARTED
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
STALLSTAT	0x400		Stall status for EasyDMA RAM accesses. The fields in this register are set to STALL by hardware whenever a stall occurres and can be cleared (set to NOSTALL) by the CPU.
ENABLE	0x500		Enable SPIM
PSEL.SCK	0x508		Pin select for SCK
PSEL.MOSI	0x50C		Pin select for MOSI signal
PSEL.MISO	0x510		Pin select for MISO signal
PSEL.CSN	0x514		Pin select for CSN
FREQUENCY	0x524		SPI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
RXD.LIST	0x540		EasyDMA list type
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
TXD.LIST	0x550		EasyDMA list type
CONFIG	0x554		Configuration register
IFTIMING.RXDELAY	0x560		Sample delay for input serial data on MISO
IFTIMING.CSNDUR	0x564		Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions
CSNPOL	0x568		Polarity of CSN output
PSELDCX	0x56C		Pin select for DCX signal
DCXCNT	0x570		DCX configuration
ORC	0x5C0		Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT

Table 147: Register overview

7.30.7.1 TASKS_START

Address offset: 0x010

Start SPI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start SPI transaction
			Trigger	1	Trigger task

7.30.7.2 TASKS_STOP

Address offset: 0x014

Stop SPI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop SPI transaction
			Trigger	1	Trigger task

7.30.7.3 TASKS_SUSPEND

Address offset: 0x01C

Suspend SPI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend SPI transaction
			Trigger	1	Trigger task

7.30.7.4 TASKS_RESUME

Address offset: 0x020

Resume SPI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume SPI transaction
			Trigger	1	Trigger task

7.30.7.5 SUBSCRIBE_START

Address offset: 0x090

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.30.7.6 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.30.7.7 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.30.7.8 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.30.7.9 EVENTS_STOPPED

Address offset: 0x104

SPI transaction has stopped

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			SPI transaction has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.30.7.10 EVENTS_ENDRX

Address offset: 0x110

End of RXD buffer reached

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			End of RXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.30.7.11 EVENTS_END

Address offset: 0x118

End of RXD buffer and TXD buffer reached

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			End of RXD buffer and TXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.30.7.12 EVENTS_ENDTX

Address offset: 0x120

End of TXD buffer reached

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			End of TXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.30.7.13 EVENTS_STARTED

Address offset: 0x14C

Transaction started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			Transaction started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.30.7.14 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.30.7.15 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.30.7.16 PUBLISH_END

Address offset: 0x198

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.30.7.17 PUBLISH_ENDTX

Address offset: 0x1A0

Publish configuration for event ENDTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.30.7.18 PUBLISH_STARTED

Address offset: 0x1CC

Publish configuration for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.30.7.19 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	END_START			Shortcut between event END and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.30.7.20 INTENSET

Address offset: 0x304

Enable interrupt

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

7.30.7.21 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.30.7.22 STALLSTAT

Address offset: 0x400

Stall status for EasyDMA RAM accesses. The fields in this register are set to STALL by hardware whenever a stall occurres and can be cleared (set to NOSTALL) by the CPU.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	TX		[1..0]	Stall status for EasyDMA RAM reads
			NOSTALL	0	No stall
			STALL	1	A stall has occurred
B	RW	RX		[1..0]	Stall status for EasyDMA RAM writes
			NOSTALL	0	No stall
			STALL	1	A stall has occurred

7.30.7.23 ENABLE

Address offset: 0x500

Enable SPIM

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPIM
			Disabled	0	Disable SPIM
			Enabled	7	Enable SPIM

7.30.7.24 PSEL.SCK

Address offset: 0x508

Pin select for SCK

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

7.30.7.25 PSEL.MOSI

Address offset: 0x50C

Pin select for MOSI signal

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

7.30.7.26 PSEL.MISO

Address offset: 0x510

Pin select for MISO signal

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

7.30.7.27 PSEL.CSN

Address offset: 0x514

Pin select for CSN

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

7.30.7.28 FREQUENCY

Address offset: 0x524

SPI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	FREQUENCY			SPI master data rate
			K125	0x02000000	125 kbps
			K250	0x04000000	250 kbps
			K500	0x08000000	500 kbps
			M1	0x10000000	1 Mbps
			M2	0x20000000	2 Mbps
			M4	0x40000000	4 Mbps
			M8	0x80000000	8 Mbps
			M16	0x0A000000	16 Mbps
			M32	0x14000000	32 Mbps

7.30.7.29 RXD.PTR

Address offset: 0x534

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.30.7.30 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

7.30.7.31 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

7.30.7.32 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.30.7.33 TXD.PTR

Address offset: 0x544

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.30.7.34 TXD.MAXCNT

Address offset: 0x548

Number of bytes in transmit buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

7.30.7.35 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

7.30.7.36 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.30.7.37 CONFIG

Address offset: 0x554

Configuration register

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

7.30.7.38 IFTIMING.RXDELAY

Address offset: 0x560

Sample delay for input serial data on MISO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RXDELAY		[7..0]	Sample delay for input serial data on MISO. The value specifies the number of 64 MHz clock cycles (15.625 ns) delay from the the sampling edge of SCK (leading edge for CONFIG.CPHA = 0, trailing edge for CONFIG.CPHA = 1) until the input serial data is sampled. As en example, if RXDELAY = 0 and CONFIG.CPHA = 0, the input serial data is sampled on the rising edge of SCK.

7.30.7.39 IFTIMING.CSNDUR

Address offset: 0x564

Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CSNDUR		[0xFF..0]	Minimum duration between edge of CSN and edge of SCK and minimum duration CSN must stay high between transactions. The value is specified in number of 64 MHz clock cycles (15.625 ns).

7.30.7.40 CSNPOL

Address offset: 0x568

Polarity of CSN output

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CSNPOL			Polarity of CSN output
			LOW	0	Active low (idle state high)
			HIGH	1	Active high (idle state low)

7.30.7.41 PSELDCX

Address offset: 0x56C

Pin select for DCX signal

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

7.30.7.42 DCXCNT

Address offset: 0x570

DCX configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DCXCNT		0x0..0xF	This register specifies the number of command bytes preceding the data bytes. The PSEL.DCX line will be low during transmission of command bytes and high during transmission of data bytes. Value 0xF indicates that all bytes are command bytes.

7.30.7.43 ORC

Address offset: 0x5C0

Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ORC			Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT.

7.30.8 Electrical specification

7.30.8.1 Timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIM}	Bit rates for SPIM[20]			16[21]	Mbps
$t_{SPIM,START}$	Time from START task to transmission started		1		µs
$t_{SPIM,CSCK}$	SCK period	125			ns

[20] High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.
[21] Application core SPIM4 supports 32 Mbps write speed when running at 128 MHz.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,RSCK,LD}$	SCK rise time, standard drive[22]			$t_{RF,25pF}$	
$t_{SPIM,RSCK,HD}$	SCK rise time, high drive[22]			$t_{HRF,25pF}$	
$t_{SPIM,FSCK,LD}$	SCK fall time, standard drive[22]			$t_{RF,25pF}$	
$t_{SPIM,FSCK,HD}$	SCK fall time, high drive[22]			$t_{HRF,25pF}$	
$t_{SPIM,WHSCK}$	SCK high time[22]	(t_{CSCK}/2) - t_{RSCK} - 1.5 ns			
$t_{SPIM,WLSCK}$	SCK low time[22]	(t_{CSCK}/2) - t_{FSCK} - 1.5 ns			
$t_{SPIM,SUMI}$	MISO to CLK edge setup time	19			ns
$t_{SPIM,HMI}$	CLK edge to MISO hold time	10			ns
$t_{SPIM,VMO}$	CLK edge to MOSI valid, SCK frequency ≤ 8 MHz			59	ns
$t_{SPIM,VMO,HS}$	CLK edge to MOSI valid, SCK frequency > 8 MHz	..	..	..	ns
$t_{SPIM,HMO}$	MOSI hold time after CLK edge	10			ns

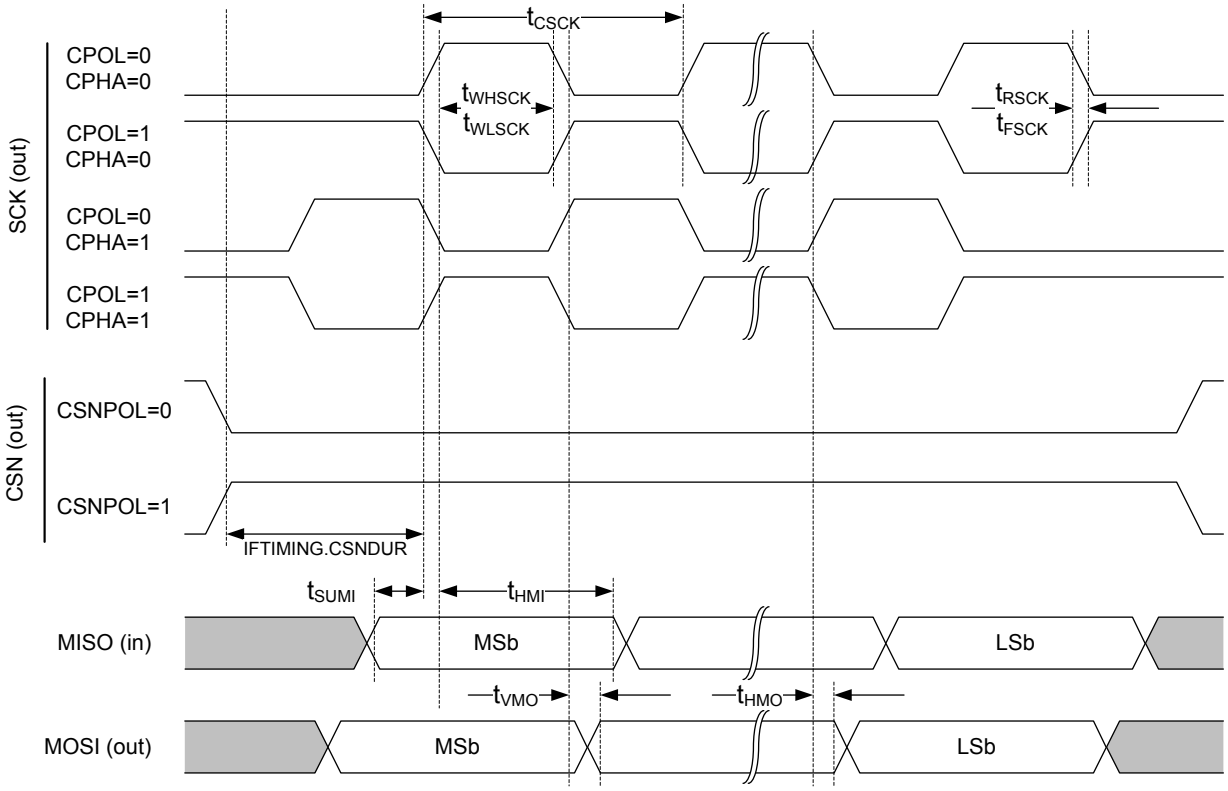

Figure 199: SPIM timing diagram

7.31 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.

[22] At 25pF load, including GPIO pin capacitance, see GPIO spec.

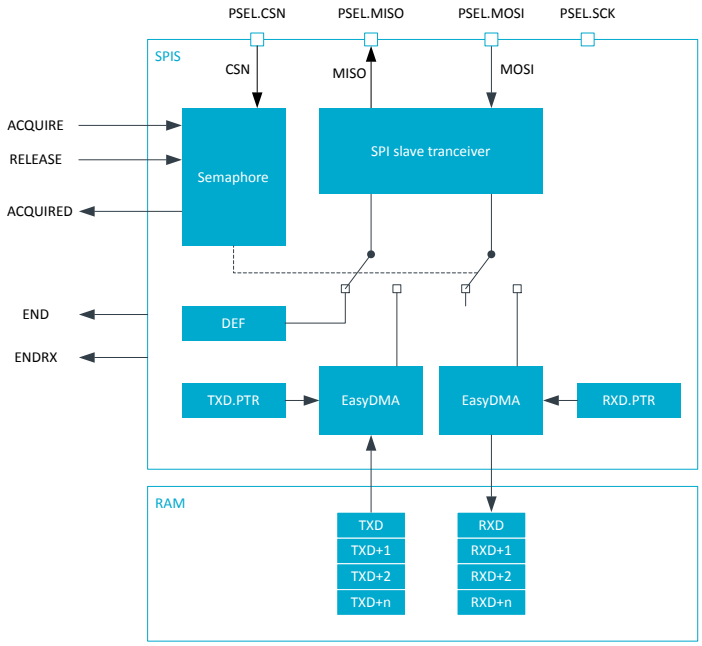

Figure 200: SPI slave

SPIS supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Active High)	0 (Trailing Edge)
SPI_MODE1	0 (Active High)	1 (Leading Edge)
SPI_MODE2	1 (Active Low)	0 (Trailing Edge)
SPI_MODE3	1 (Active Low)	1 (Leading Edge)

Table 148: SPI modes

7.31.1 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 Peripherals on page 145 shows which peripherals have the same ID as the SPI slave.

7.31.2 EasyDMA

SPIS implements EasyDMA for accessing RAM without CPU involvement.

SPIS implements the EasyDMA channels found in the following table.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 149: SPIS EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 149.

If RXD.MAXCNT is larger than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register.

The END event indicates that EasyDMA has finished accessing the buffer in RAM.

7.31.3 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.

Before the CPU can safely update the RXD.PTR and TXD.PTR pointers it must first acquire the SPI semaphore. The CPU can acquire the semaphore by triggering the ACQUIRE task and then receiving the ACQUIRED event. When the CPU has updated the RXD.PTR and TXD.PTR pointers the CPU must release the semaphore before the SPI slave will be able to acquire it.

The CPU releases the semaphore by triggering the RELEASE task, this is illustrated in SPI transaction when shortcut between END and ACQUIRE is enabled on page 568. Triggering the RELEASE task when the semaphore is not granted to the CPU will have no effect. See Semaphore operation on page 568 for more information

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.

The ENDRX event is generated when the RX buffer has been filled.

The RXD.MAXCNT register specifies the maximum number of bytes the SPI slave can receive in one granted transaction. If the SPI slave receives more than RXD.MAXCNT number of bytes, an OVERFLOW will be indicated in the STATUS register and the incoming bytes will be discarded.

The TXD.MAXCNT 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 TXD.MAXCNT 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.

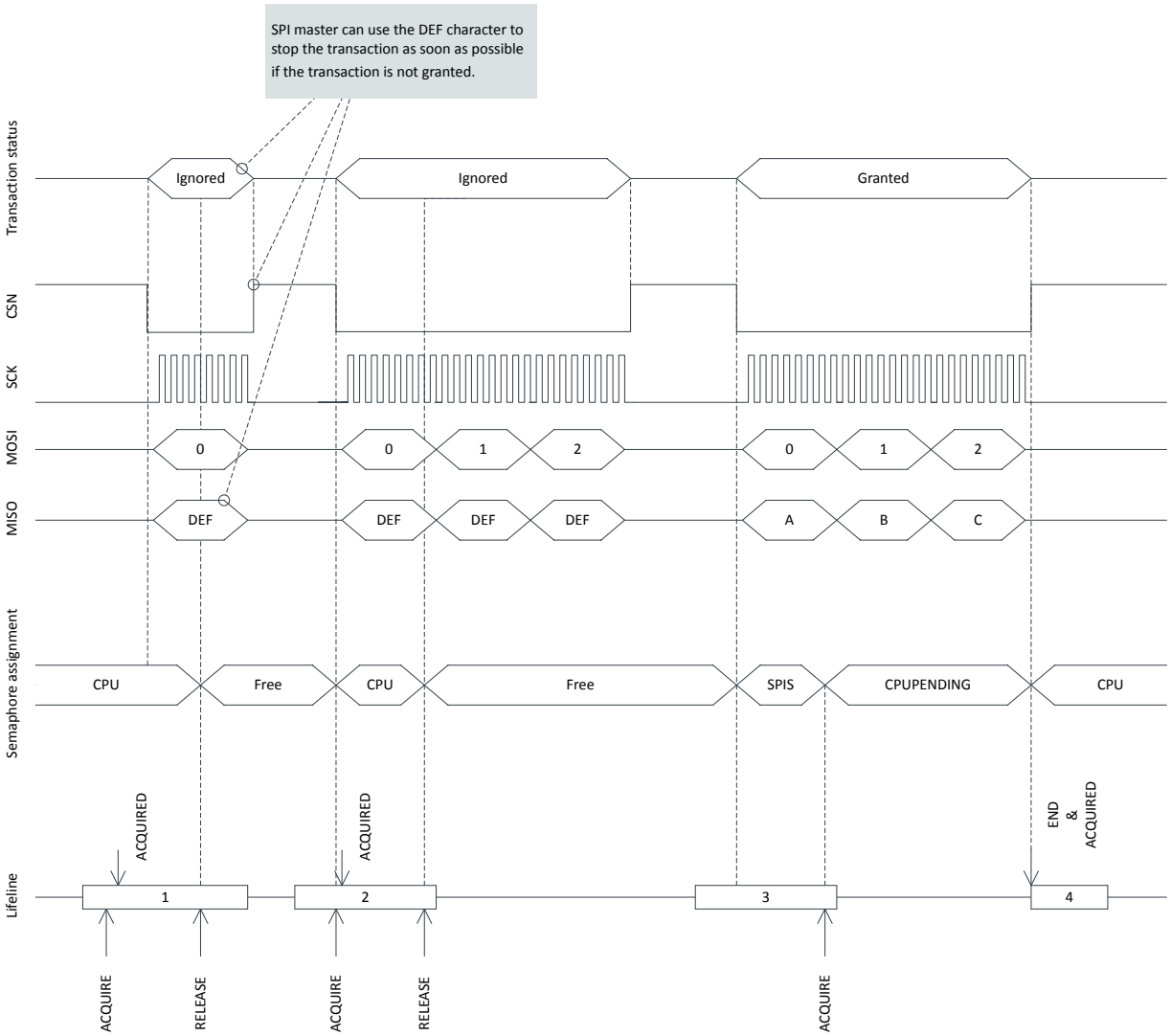

Figure 201: SPI transaction when shortcut between END and ACQUIRE is enabled

7.31.4 Semaphore operation

The semaphore is a mechanism implemented inside the SPI slave that prevents simultaneous access to the data buffers by the SPI slave and CPU.

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 figure SPI semaphore FSM on page 569 illustrates the transitions between states in the semaphore based on the relevant tasks and events.

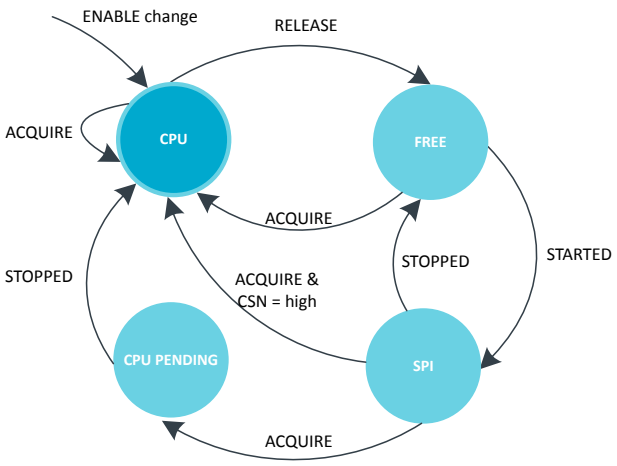

Figure 202: SPI semaphore FSM

Note: 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 SPI slave will try to acquire the semaphore when STARTED event is detected, the even also indicates that CSN is currently low. If the SPI slave does not manage to acquire the semaphore at this point (i.e., it is under CPU's control), 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 SPI transaction when shortcut between END and ACQUIRE is enabled on page 568, 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 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.

7.31.5 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 control on page 43 chapter for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field

and PIN_CNF[n] register. PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI slave is disabled.

To secure correct behavior in the SPI slave, the pins used by the SPI slave must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 570 before enabling the SPI slave. This is to secure that the pins used by the SPI slave are driven correctly if the SPI slave itself is temporarily disabled, or if the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the SPI slave is to be recognized by an external SPI master.

The MISO line is set in high impedance as long as the SPI slave is not selected with CSN.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI signal	SPI pin	Direction	Output value	Comment
CSN	As specified in PSEL.CSN	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
MOSI	As specified in PSEL.MOSI	Input	Not applicable	
MISO	As specified in PSEL.MISO	Input	Not applicable	Emulates that the SPI slave is not selected.

Table 150: GPIO configuration before enabling peripheral

7.31.6 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	APPLICATION	SPIS	SPIS0 : S SPIS0 : NS	US	SA	SPI slave 0	
0x50009000 0x40009000	APPLICATION	SPIS	SPIS1 : S SPIS1 : NS	US	SA	SPI slave 1	
0x5000B000 0x4000B000	APPLICATION	SPIS	SPIS2 : S SPIS2 : NS	US	SA	SPI slave 2	
0x5000C000 0x4000C000	APPLICATION	SPIS	SPIS3 : S SPIS3 : NS	US	SA	SPI slave 3	
0x41013000	NETWORK	SPIS	SPIS0	NS	NA	SPI slave 0	

Table 151: Instances

Register	Offset	Security	Description
TASKS_ACQUIRE	0x024		Acquire SPI semaphore
TASKS_RELEASE	0x028		Release SPI semaphore, enabling the SPI slave to acquire it
SUBSCRIBE_ACQUIRE	0x0A4		Subscribe configuration for task ACQUIRE
SUBSCRIBE_RELEASE	0x0A8		Subscribe configuration for task RELEASE
EVENTS_END	0x104		Granted transaction completed
EVENTS_ENDRX	0x110		End of RXD buffer reached
EVENTS_ACQUIRED	0x128		Semaphore acquired
PUBLISH_END	0x184		Publish configuration for event END
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_ACQUIRED	0x1A8		Publish configuration for event ACQUIRED
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
SEMSTAT	0x400		Semaphore status register
STATUS	0x440		Status from last transaction
ENABLE	0x500		Enable SPI slave

Register	Offset	Security	Description	
PSEL.SCK	0x508		Pin select for SCK	
PSEL.MISO	0x50C		Pin select for MISO signal	
PSEL.MOSI	0x510		Pin select for MOSI signal	
PSEL.CSN	0x514		Pin select for CSN signal	
PSELSCK	0x508		Pin select for SCK	Deprecated
PSELMISO	0x50C		Pin select for MISO	Deprecated
PSELMOSI	0x510		Pin select for MOSI	Deprecated
PSELCSN	0x514		Pin select for CSN	Deprecated
RXD.PTR	0x534		RXD data pointer	
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer	
RXD.AMOUNT	0x53C		Number of bytes received in last granted transaction	
RXD.LIST	0x540		EasyDMA list type	
RXDPTR	0x534		RXD data pointer	Deprecated
MAXRX	0x538		Maximum number of bytes in receive buffer	Deprecated
AMOUNTRX	0x53C		Number of bytes received in last granted transaction	Deprecated
TXD.PTR	0x544		TXD data pointer	
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer	
TXD.AMOUNT	0x54C		Number of bytes transmitted in last granted transaction	
TXD.LIST	0x550		EasyDMA list type	
TXDPTR	0x544		TXD data pointer	Deprecated
MAXTX	0x548		Maximum number of bytes in transmit buffer	Deprecated
AMOUNTTX	0x54C		Number of bytes transmitted in last granted transaction	Deprecated
CONFIG	0x554		Configuration register	
DEF	0x55C		Default character. Character clocked out in case of an ignored transaction.	
ORC	0x5C0		Over-read character	

Table 152: Register overview

7.31.6.1 TASKS_ACQUIRE

Address offset: 0x024

Acquire SPI semaphore

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_ACQUIRE			Acquire SPI semaphore
			Trigger	1	Trigger task

7.31.6.2 TASKS_RELEASE

Address offset: 0x028

Release SPI semaphore, enabling the SPI slave to acquire it

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_RELEASE			Release SPI semaphore, enabling the SPI slave to acquire it
			Trigger	1	Trigger task

7.31.6.3 SUBSCRIBE_ACQUIRE

Address offset: 0x0A4

Subscribe configuration for task ACQUIRE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task ACQUIRE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.31.6.4 SUBSCRIBE_RELEASE

Address offset: 0x0A8

Subscribe configuration for task RELEASE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RELEASE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.31.6.5 EVENTS_END

Address offset: 0x104

Granted transaction completed

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_END			Granted transaction completed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.31.6.6 EVENTS_ENDRX

Address offset: 0x110

End of RXD buffer reached

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			End of RXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.31.6.7 EVENTS_ACQUIRED

Address offset: 0x128

Semaphore acquired

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ACQUIRED			Semaphore acquired
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.31.6.8 PUBLISH_END

Address offset: 0x184

Publish configuration for event END

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.31.6.9 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.31.6.10 PUBLISH_ACQUIRED

Address offset: 0x1A8

Publish configuration for event ACQUIRED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ACQUIRED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.31.6.11 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	END_ACQUIRE			Shortcut between event END and task ACQUIRE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.31.6.12 INTENSET

Address offset: 0x304

Enable interrupt

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

7.31.6.13 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACQUIRED			Write '1' to disable interrupt for event ACQUIRED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.31.6.14 SEMSTAT

Address offset: 0x400

Semaphore status register

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

ID	R/W	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

7.31.6.15 STATUS

Address offset: 0x440

Status from last transaction

Note: 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	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'

7.31.6.16 ENABLE

Address offset: 0x500

Enable SPI slave

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPI slave
			Disabled	0	Disable SPI slave
			Enabled	2	Enable SPI slave

7.31.6.17 PSEL.SCK

Address offset: 0x508

Pin select for SCK

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.31.6.18 PSEL.MISO

Address offset: 0x50C

Pin select for MISO signal

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.31.6.19 PSEL.MOSI

Address offset: 0x510

Pin select for MOSI signal

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

7.31.6.20 PSEL.CSN

Address offset: 0x514

Pin select for CSN signal

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

7.31.6.21 PSELSCK (Deprecated)

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PSELSCK		[0..31]	Pin number configuration for SPI SCK signal
			Disconnected	0xFFFFFFFF	Disconnect

7.31.6.22 PSELMISO (Deprecated)

Address offset: 0x50C

Pin select for MISO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PSELMISO		[0..31]	Pin number configuration for SPI MISO signal
			Disconnected	0xFFFFFFFF	Disconnect

7.31.6.23 PSELMOSI (Deprecated)

Address offset: 0x510

Pin select for MOSI

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PSELMOSI		[0..31]	Pin number configuration for SPI MOSI signal
			Disconnected	0xFFFFFFFF	Disconnect

7.31.6.24 PSELCSN (Deprecated)

Address offset: 0x514

Pin select for CSN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PSELCSN		[0..31]	Pin number configuration for SPI CSN signal
			Disconnected	0xFFFFFFFF	Disconnect

7.31.6.25 RXD.PTR

Address offset: 0x534

RXD data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			RXD data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.31.6.26 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

7.31.6.27 RXD.AMOUNT

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes received in the last granted transaction

7.31.6.28 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.31.6.29 RXDPTR (Deprecated)

Address offset: 0x534

RXD data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RXDPTR			RXD data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.31.6.30 MAXRX (Deprecated)

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXRX		[1..0xFFFF]	Maximum number of bytes in receive buffer

7.31.6.31 AMOUNTRX (Deprecated)

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNTRX		[1..0xFFFF]	Number of bytes received in the last granted transaction

7.31.6.32 TXD.PTR

Address offset: 0x544

TXD data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			TXD data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.31.6.33 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

7.31.6.34 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transmitted in last granted transaction

7.31.6.35 TXD.LIST

Address offset: 0x550

EasyDMA list type

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.31.6.36 TXDPTR (Deprecated)

Address offset: 0x544

TXD data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	TXDPTR			TXD data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.31.6.37 MAXTX (Deprecated)

Address offset: 0x548

Maximum number of bytes in transmit buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXTX		[1..0xFFFF]	Maximum number of bytes in transmit buffer

7.31.6.38 AMOUNTTX (Deprecated)

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNTTX		[1..0xFFFF]	Number of bytes transmitted in last granted transaction

7.31.6.39 CONFIG

Address offset: 0x554

Configuration register

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

ID	R/W	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

7.31.6.40 DEF

Address offset: 0x55C

Default character. Character clocked out in case of an ignored transaction.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	DEF			Default character. Character clocked out in case of an ignored transaction.

7.31.6.41 ORC

Address offset: 0x5C0

Over-read character

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character clocked out after an over-read of the transmit buffer.

7.31.7 Electrical specification

7.31.7.1 SPIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIS}	Bit rates for SPIS[23]			8[24]	Mbps
$t_{SPIS,START}$	Time from RELEASE task to receive/transmit (CSN active)	..	..	..	µs

[23] High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

[24] The actual maximum data rate depends on the master's CLK to MISO and MOSI setup and hold timings.

7.31.7.2 Serial Peripheral Interface Slave (SPIS) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,CSCKIN}$	SCK input period	125			ns
$t_{SPIS,RFSCKIN}$	SCK input rise/fall time			30	ns
$t_{SPIS,WHSCKIN}$	SCK input high time	30			ns
$t_{SPIS,WLSCKIN}$	SCK input low time	30			ns
$t_{SPIS,SUCSN}$	CSN to CLK setup time	1000[25]			ns
$t_{SPIS,HCSN}$	CLK to CSN hold time	1000			ns
$t_{SPIS,ASA}$	CSN to MISO driven			68	ns
$t_{SPIS,ASO}$	CSN to MISO valid[26]	1000			ns
$t_{SPIS,DISSO}$	CSN to MISO disabled[26]			68	ns
$t_{SPIS,CWH}$	CSN inactive time	300			ns
$t_{SPIS,VSO}$	CLK edge to MISO valid			31	ns
$t_{SPIS,HSO}$	MISO hold time after CLK edge	13			ns
$t_{SPIS,SUSI}$	MOSI to CLK edge setup time	19			ns
$t_{SPIS,HSI}$	CLK edge to MOSI hold time	10			ns

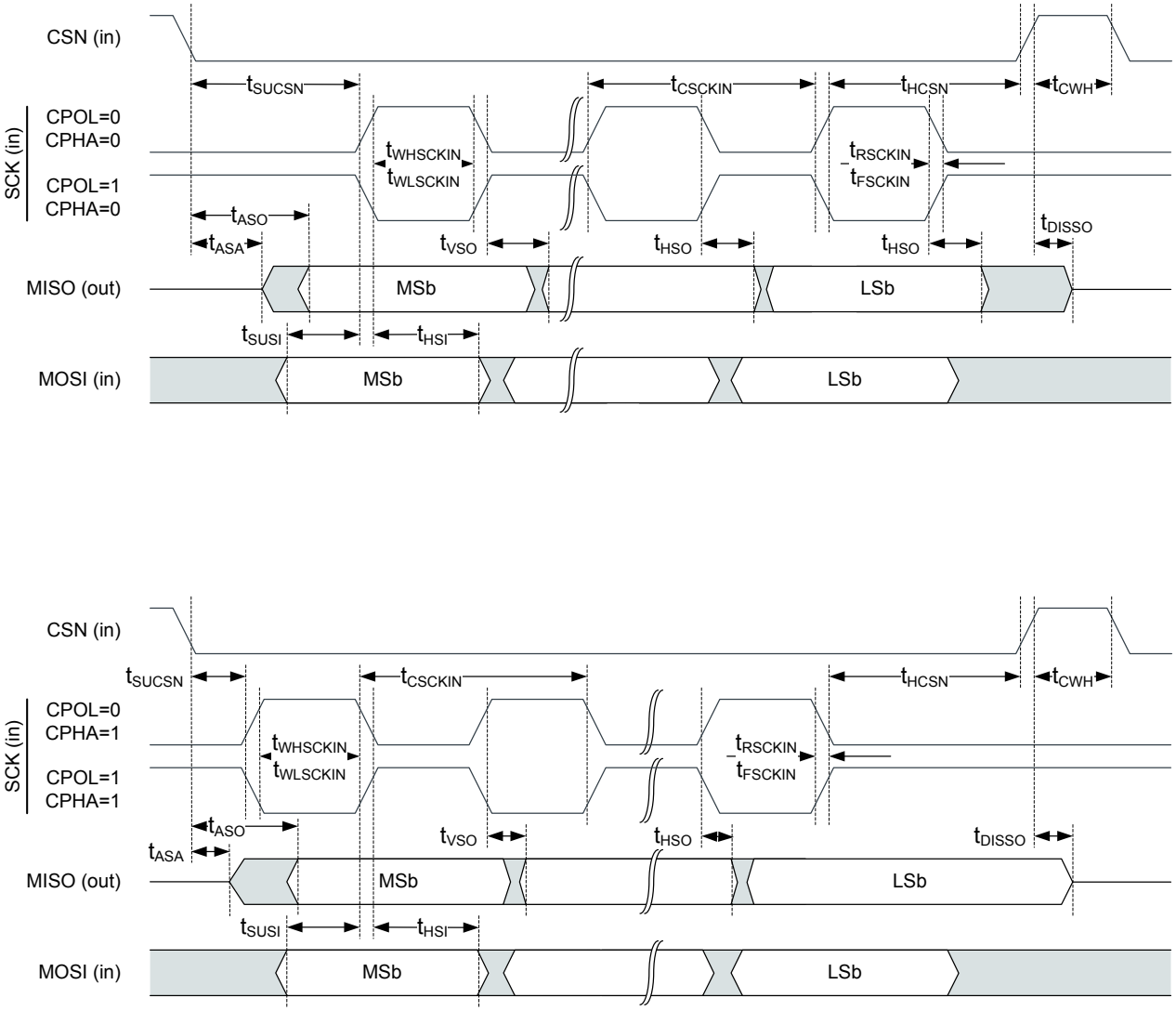

Figure 203: SPIS timing diagram

[25] Excluding any start-up delay for the high frequency clock in low power mode.
[26] At 25pF load, including GPIO capacitance, see GPIO spec.

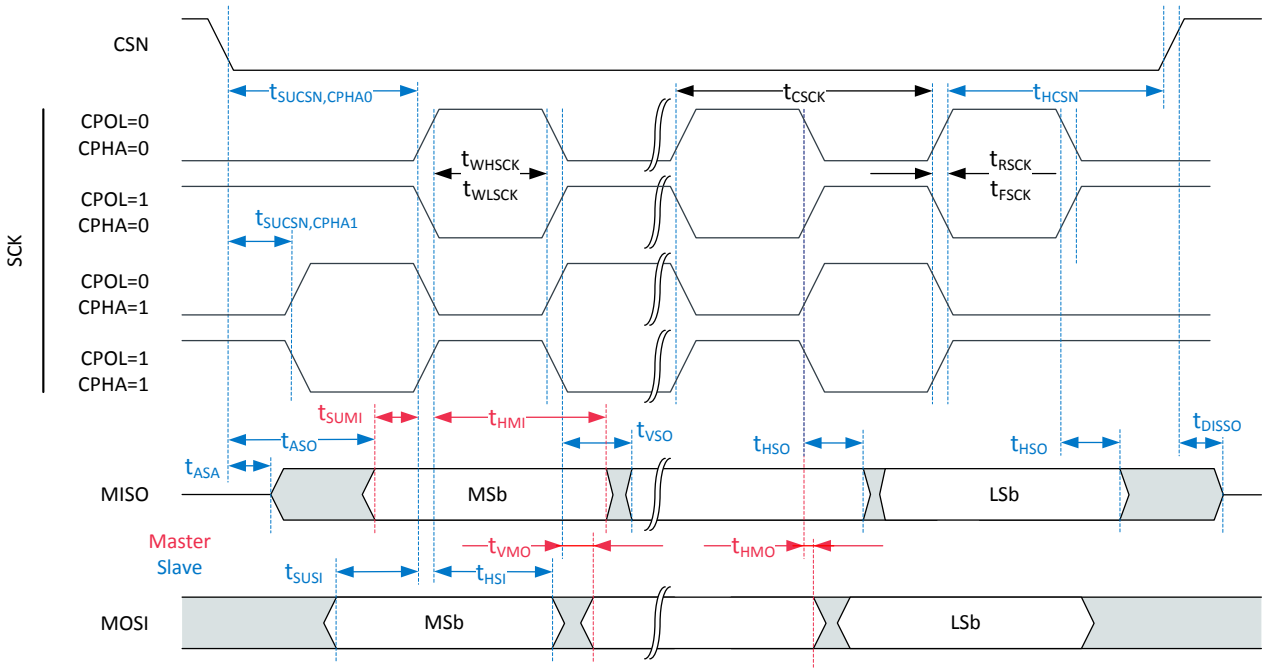

Figure 204: Common SPIM and SPIS timing diagram

7.32 SPU — System protection unit

SPU is the central point in the system to control access to memories, peripherals and other resources.

The main features of SPU are the following:

- Arm TrustZone support, allowing definition of secure, non-secure and non-secure callable memory regions
- Extended Arm TrustZone, protecting memory regions and peripherals from non-CPU devices like EasyDMA transfer
- Pin access protection, preventing non-secure code and peripherals from accessing secure pin resources
- DPPI access protection, realized by preventing non-secure code and peripherals to publish from or subscribe to secured DPPI channels
- External domain access protection, controlling access rights from other MCUs

7.32.1 General concepts

SPU provides a register interface to control the various internal logic blocks that monitor access to memory-mapped slave devices (RAM, flash, peripherals, etc) and other resources (device pins, DPPI channels, etc).

For memory-mapped devices like RAM, flash, and peripherals, the internal logic checks the address and attributes (e.g. read, write, execute, secure) of the incoming transfer to block it if necessary. A secure resource can be accessed by a given master based on the following factors:

- *CPU-type master* – By the security state of the CPU and the security state reported by SPU, for the address in the bus transfer.
- *Non-CPU master* – By the security attribute of the master that initiates the transfer, defined by a SPU register.

The Simplified view of SPU protection on page 585 shows a simplified view of the SPU registers controlling several internal modules.

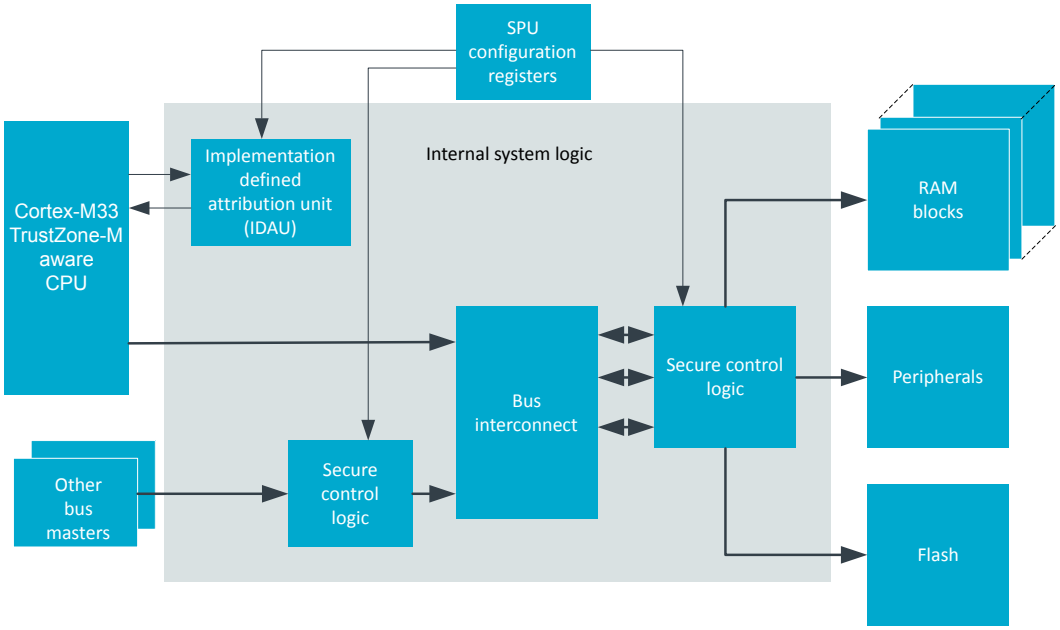

Figure 205: Simplified view of SPU protection

The protection logic implements a read-as-zero/write-ignore (RAZ/WI) policy where the following are true:

- A blocked read operation will always return a zero value on the bus, preventing information leak.
- A write operation to a forbidden region or peripheral will be ignored.

An error is reported through dedicated error signals. For security state violations from an M33 master this will be a SecureFault exception, for other violations this will be an SPU event. The SPU event can be configured to generate an interrupt towards the CPU.

Other resources like pins and DPPI channels are protected by comparing the security attributes of the protected resource with the security attribute of the peripheral that wants to access it. SPU is the only place where those security attributes can be configured.

7.32.1.1 Special considerations for Arm TrustZone for Cortex-M enabled system

SPU also controls custom logic for an Arm TrustZone for Cortex-M enabled CPU.

Custom logic is shown as the implementation defined attribution unit (IDAU) in figure Simplified view of SPU protection on page 585. Full support is provided for the following:

- Arm TrustZone for Cortex-M related instructions, like test target (TT) for reporting the security attributes of a region
- Non-secure callable (NSC) regions, to implement secure entry points from non-secure code

SPU provides the necessary registers to configure the security attributes for memory regions and peripherals. However, as a requirement to use SPU, the secure attribution unit (SAU) needs to be disabled and all memory set as non-secure in the Arm core. This will allow SPU to control the IDAU and set the security attribution of all addresses as originally intended.

7.32.2 Flash access control

The flash memory space is divided into 64 regions of 16 KiB, each with configurable permissions settings.

For each region, the following types of permissions can be configured:

Read

Allows data read access to the region. The code fetch from this region is not controlled by the read permission but by the execute permission described below.

Write
Allows write or page erase access to the region.

Execute
Allows code fetch from this region, even if data read is disabled.

Secure
Allows only bus accesses with the security attribute set to access the region.

Permissions can be set independently. For example, it is possible to configure a flash region to be accessible only through secure transfer, being read-only (no write allowed) and non-executable (no code fetch allowed). For each region, permissions can be set and then locked by using the FLASHREGION[n].PERM.LOCK bit, to prevent subsequent modifications.

The debugger can step through execute-protected memory regions.

The following figure shows the N=64 flash regions, each of size 16 KiB.

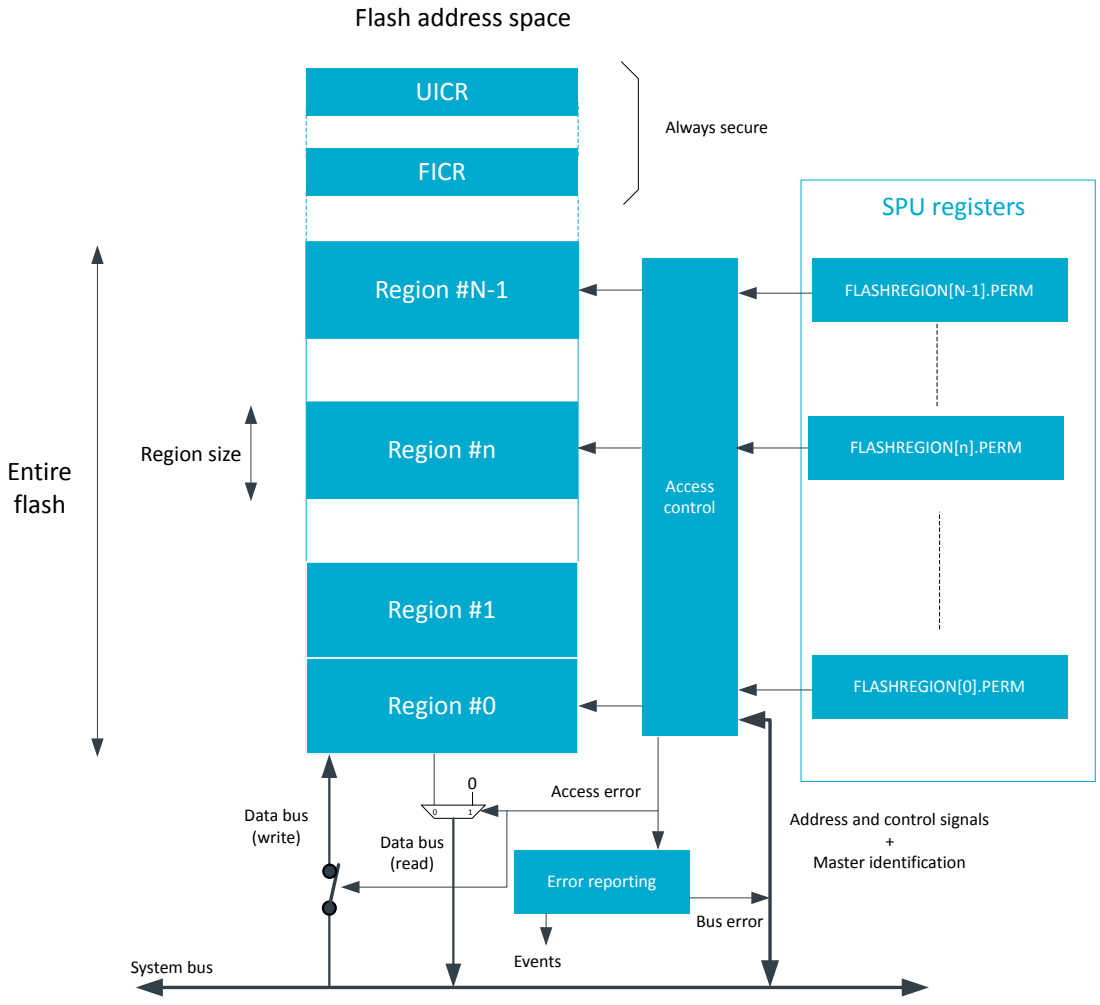

Figure 206: Flash memory regions

7.32.2.1 Non-secure callable (NSC) region definition in flash

SPU provides support for the definition of non-secure callable (NSC) sub-regions to allow non-secure to secure function calls.

A non-secure callable sub-region can only exist within an existing secure region and its definition is done using the following registers:

- FLASHNSC[n].REGION, used to select the secure region that will contain the NSC sub-region
- FLASHNSC[n].SIZE, used to define the size of the NSC sub-region within the secure region

The NSC sub-region will be defined starting with the highest address in that region and descending. The following figure illustrates the NSC sub-regions and the registers used for their definition.

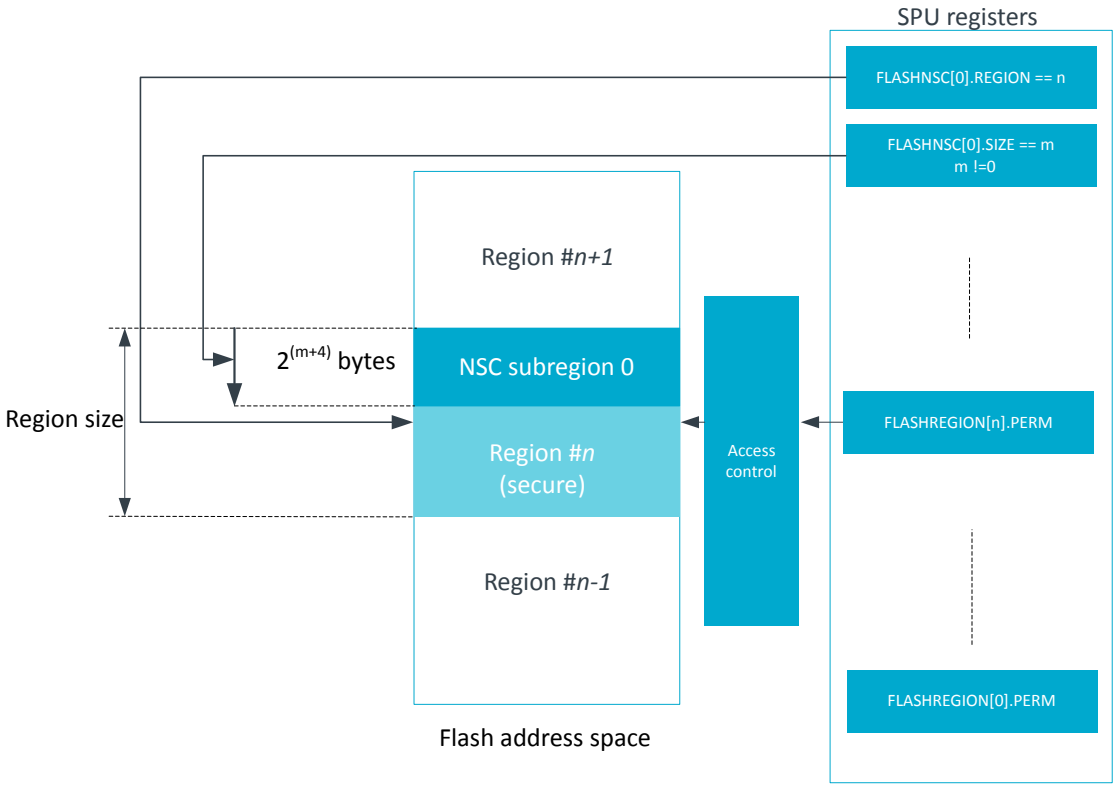

Figure 207: Non-secure callable region definition in the flash memory space

The NSC sub-region will only be defined when the following are true:

- FLASHNSC[i].SIZE value is non zero
- FLASHNSC[i].REGION defines a secure region

If FLASHNSC[i].REGION and FLASHNSCj].REGION have the same value, there is only one sub-region defined as NSC, with the size given by the maximum of FLASHNSC[i].SIZE and FLASHNSC[j].SIZE.

If FLASHNSC[i].REGION defines a non-secure region, then there is no non-secure callable region defined and the selected region stays non-secure.

7.32.2.2 Flash access error reporting

SPU and the logic controlled by it will respond with a certain behavior once an access violation is detected.

The following actions will happen once the logic controlled by SPU detects an access violation on one of the flash ports:

- The faulty transfer will be blocked.
- In case of a read transfer, the bus will be driven to zero.
- If supported by the master, feedback will be sent to the master through specific bus error signals. At the same time, SPU will receive an event that can optionally trigger a CPU interrupt.
- A SecureFault exception will be triggered if a security violation is detected for access from the CPU.
- A BusFault exception will be triggered when a read/write/execute protection violation is detected from the CPU.
- The FLASHACCERR event will be triggered if any access violations are detected for all master types except for the CPU security violation.

The following table summarizes the SPU behavior based on the type of initiator and access violation.

Master type	Security violation	Read/Write/Execute protection violation
Arm Cortex-M33	SecureFault exception	BusFault exception, FLASHACCERR event
EasyDMA	RAZ/WI, FLASHACCERR event	RAZ/WI, FLASHACCERR event
Other masters	RAZ/WI, FLASHACCERR event	RAZ/WI, FLASHACCERR event

Table 153: Error reporting for flash access errors

For the Arm Cortex-M33 master, the SecureFault exception will take precedence over the BusFault exception if a security violation occurs simultaneously with another type of violation.

7.32.2.3 UICR and FICR protections

The user information configuration registers (UICR) and factory information configuration registers (FICR) are always considered as secure. FICR registers are read-only. UICR registers can be read and written by secure code only.

Writing new values to user information configuration registers must follow the procedure described in NVMC — Non-volatile memory controller on page 329. Code execution from FICR and UICR address spaces will always be reported as an access violation except during a debug session.

7.32.3 RAM access control

The RAM memory space is divided into 64 regions of 8 KiB, each with configurable permissions settings.

For each region, the following types of permissions can be configured:

Read
Allows data read access to the region. Code fetch from this region is not controlled by the read permission but by the execute permission described below.

Write
Allows write access to the region.

Execute
Allows code fetch from this region.

Secure
Allows only bus accesses with the security attribute set to access the region.

Permissions can be set independently. For example, it is possible to configure a RAM region to be accessible only through secure transfer, being read-only (no write allowed) and non-executable (no code fetch allowed). For each region, permissions can be set and then locked to prevent subsequent modifications by using the RAMREGION[n].PERM.LOCK bit.

The following figure shows the RAM memory space divided into N=64 regions, each of 8 KiB.

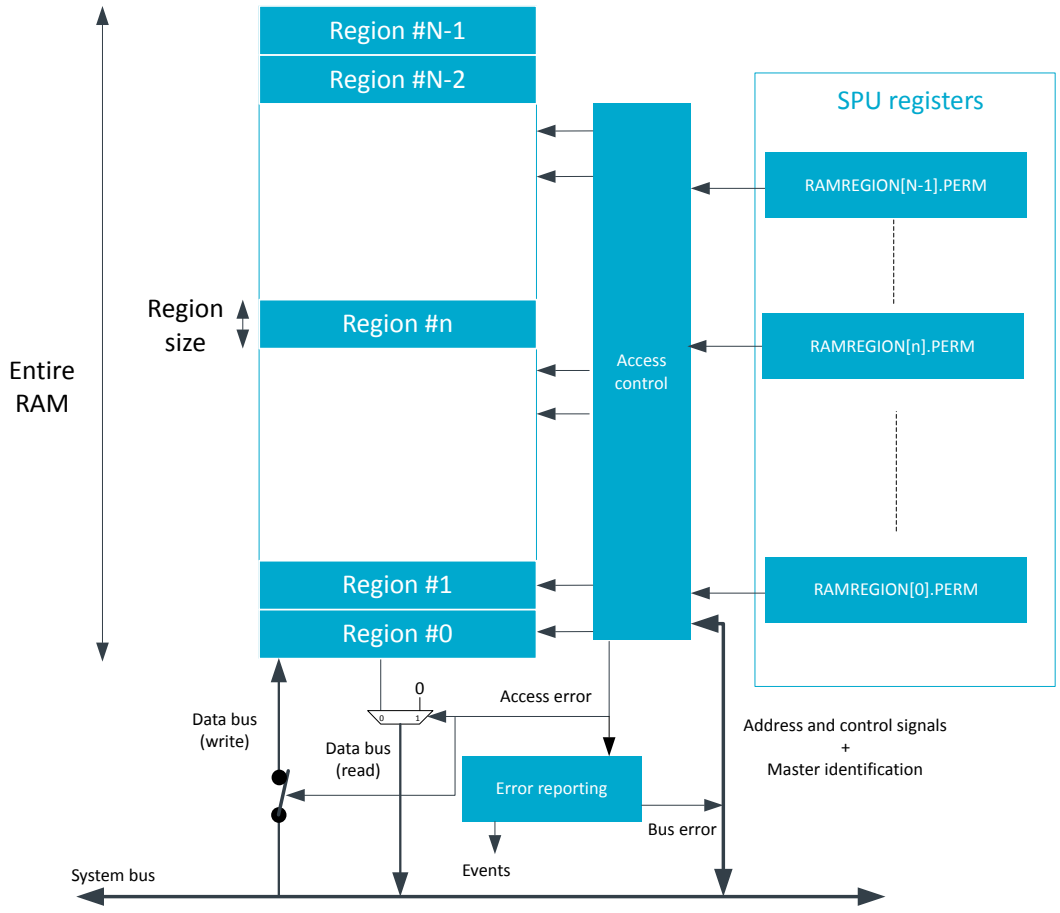

Figure 208: RAM memory regions

7.32.3.1 Non-secure callable (NSC) region definition in RAM

SPU provides support for the definition of non-secure callable (NSC) sub-regions to allow non-secure to secure function calls.

A non-secure callable sub-region can only exist within an existing secure region. It is defined by the following registers:

- RAMNSC[n].REGION, used to select the secure region that will contain the NSC sub-region
- RAMNSC[n].SIZE, used to define the size of the NSC sub-region within the secure region

The NSC sub-region will be defined starting with the highest address in that region and descending. The following figure shows the NSC sub-regions and the registers used for their definition.

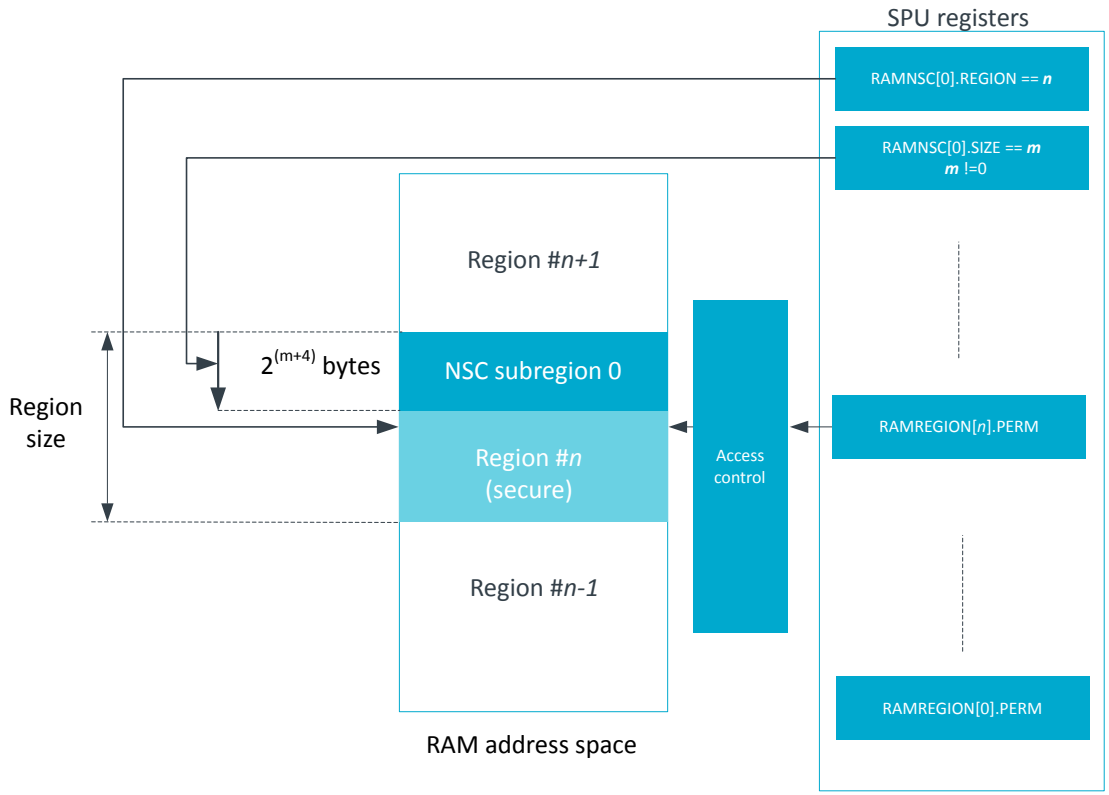

Figure 209: Non-secure callable region definition in the RAM memory space

The NSC sub-region will only be defined when the following are true:

- RAMNSC[*i*].SIZE value is non zero
- RAMNSC[*i*].REGION defines a secure region

If RAMNSC[*i*].REGION and RAMNSC[*j*].REGION have the same value, there is only one sub-region defined as NSC, with the size given by the maximum of RAMNSC[*i*].SIZE and RAMNSC[*j*].SIZE.

If RAMNSC[*i*].REGION defines a non-secure region, then there is no non-secure callable region defined and the selected region stays non-secure.

7.32.3.2 RAM access error reporting

SPU and the logic it controls will respond with a certain behavior once an access violation is detected.

The following actions will happen once the logic controlled by the SPU detects an access violation on one of the RAM ports:

- The faulty transfer will be blocked.
- In case of a read transfer, the bus will be driven to zero.
- If supported by the master, feedback will be sent to the master through specific bus error signals.
- A SecureFault exception will be triggered if security violation is detected for access from Arm Cortex-M33
- A BusFault exception will be triggered when read/write/execute protection violation is detected for Arm Cortex-M33. The SPU will also generate an event that can optionally trigger an interrupt towards the CPU.
- The RAMACCERR event will be triggered if any access violations are detected for all master types but for Arm Cortex-M33 security violation

The following table summarizes the SPU behavior based on the type of initiator and access violation.

Master type	Security violation	Read/Write/Execute protection violation
Arm Cortex-M33	SecureFault exception	BusFault exception, RAMACCERR event
EasyDMA	RAZ/WI, RAMACCERR event	RAZ/WI, RAMACCERR event
Other masters	RAZ/WI, RAMACCERR event	RAZ/WI, RAMACCERR event

Table 154: Error reporting for RAM access errors

For the Arm Cortex-M33 master, the SecureFault exception will take precedence over the BusFault exception if a security violation occurs simultaneously with another type of violation.

7.32.4 Peripheral access control

Access controls are defined by the characteristics of the peripheral.

Peripherals can have their security attribute set as one of the following:

Always secure
For a peripheral related to system control.

Always non-secure
For some general-purpose peripherals.

Configurable
For general-purpose peripherals that may be configured for secure only access.

The full list of peripherals and their corresponding security attributes can be found in Memory on page 18. For each peripheral with ID *n*, PERIPHID[*n*]. PERM will show whether the security attribute for this peripheral is configurable or not.

If not hardcoded, the security attribute can configured using the PERIPHID[*id*].PERM.

At reset, all user-selectable and split security peripherals are set to be secure with secure DMA where present.

Secure code can access both secure peripherals and non-secure peripherals.

7.32.4.1 Peripherals with split security

Peripherals with split security are defined to handle use-cases when both secure and non-secure code needs to control the same resource.

When peripherals with split security have their security attribute set to non-secure, access to specific registers and bitfields within some registers is dependent on the security attribute of the bus transfer. For example, some registers will not be accessible for a non-secure transfer.

When peripherals with split security have their security attribute set to secure, then only secure transfers can access their registers.

See Peripherals on page 145 for an overview of split security peripherals. Respective peripheral chapters explain the specific security behavior of each peripheral.

7.32.4.2 Peripheral address mapping

Peripherals that have non-secure security mapping have their address starting with 0x4XXXXXXX. Peripherals that have secure security mapping have their address starting with 0x5XXXXXXX.

Peripherals with a user-selectable security mapping are available at an address starting with the following:

- 0x4XXXXXXX, if the peripheral security attribute is set to non-secure
- 0x5XXXXXXX, if the peripheral security attribute is set to secure

Note:

Access to a secure peripheral using the 0x4XXXXXXX address range will result in bus error, regardless if the CPU is executing secure or non-secure code.

Similarly, a CPU running secure code attempting to access a non-secure peripheral using the 0x5XXXXXXX address range will result in bus error.

Peripherals with a split security mapping are available at an address starting with the following:

- 0x4XXXXXXX for non-secure access and 0x5XXXXXXX for secure access, if the peripheral security attribute is set to non-secure
  - Secure registers in the 0x4XXXXXXX range are not visible for secure or non-secure code, and an attempt to access such a register will result in write-ignore, read-as-zero behavior
  - Secure code can access both non-secure and secure registers in the 0x5XXXXXXX range
- 0x5XXXXXXX, if the peripheral security attribute is set to secure

Any attempt to access the 0x50000000 to 0x5FFFFFFF address range from non-secure code will be ignored and generate a SecureFault exception.

The following table contains the address mapping for the three peripheral types in each configuration.

Security-features and configuration	Mapped at 0x4XXXXXXX?	Mapped at 0x5XXXXXXX?
Secure peripheral	No	Yes
Non-secure peripheral	Yes	No
Split-security peripheral, with attribute=secure	No	Yes
Split-security peripheral, with attribute=non-secure	Yes, restricted functionality	Yes

Table 155: Peripheral's address mapping in relation to its security-features and configuration

7.32.4.3 Special considerations for peripherals with DMA master

Peripherals containing a DMA master can be configured so the security attribute of the DMA transfers is different from the security attribute of the peripheral itself. This allows a secure peripheral to do non-secure data transfers to or from the system memories.

The following conditions must be met:

- The DMA field of PERIPHID[n].PERM.SECURITYMAPPING should read as "SeparateAttribute"
- The peripheral itself should be secure (PERIPHID[n].PERM.SECATTR == 1)

Then it is possible to select the security attribute of the DMA transfers using the field DMASEC (PERIPHID[n].PERM.DMASEC == Secure and PERIPHID[n].PERM.DMASEC == NonSecure) in PERIPHID[n].PERM.

7.32.4.4 Peripheral access error reporting

Peripherals send error reports once access violation is detected.

The following actions will happen if the logic controlled by the SPU detects an access violation on one of the peripherals:

- The faulty transfer will be blocked.
- In case of a read transfer, the bus will be driven to zero.
- If supported by the master, feedback is sent to the master through specific bus error signals. If the master is a processor supporting Arm TrustZone for Cortex-M, a SecureFault exception will be generated for security related errors.
- The PERIPHACCERR event will be triggered.

7.32.5 Pin access control

Access to device pins can be controlled by SPU. A pin can be declared as secure so that only secure peripherals or secure code can access it. Pins declared as non-secure can be accessed by both secure and non-secure peripherals or code.

The security attribute of each pin can be individually configured in SPU's GPIOPORT[n].PERM register. When the secure attribute is set for a pin, only peripherals that have the secure attribute set will be able to read the value of the pin or change it.

Peripherals can select the pins they need access to through their PSEL registers. If a peripheral has its attribute set to non-secure, but one of its PSEL registers selects a pin with the attribute set to secure, the SPU controlled logic will ensure that the pin selection is not propagated. In addition, the pin value will always be read as zero, to prevent a non-secure peripheral from obtaining a value from a secure pin. Access to other pins with attribute set as non-secure will not be blocked.

Pins can be assigned to other domains than the application domain by changing the MCUSEL value in the GPIO PIN_CNF[n] register. Domains that do not have a pin assigned to them cannot control a pin or read its status. Any pin configuration set in a domain that doesn't have ownership of that pin will not take effect until the MCUSEL is updated to assign that pin to the domain. Within each domain, pin access is controlled by that domain's local security configuration and peripheral PSEL registers. This is illustrated in the following figure:

The SPU setting will still count when the APP domain accesses its local GPIO peripheral, as local registers are still writable even though MCUSEL is set to a different domain. Any changes in the APP GPIO peripheral done to a GPIO controlled by another domain will not affect the GPIO pad until MCUSEL is changed to APP.

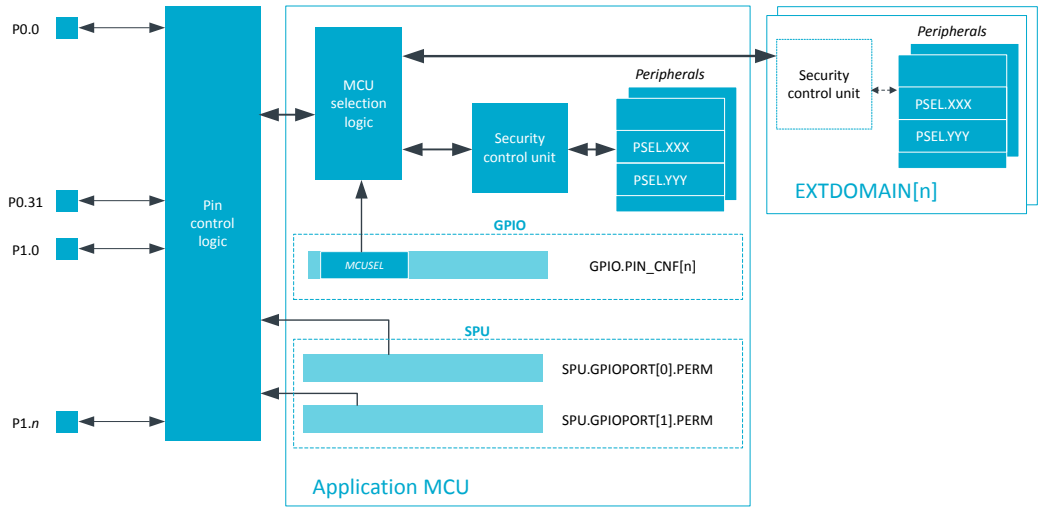

Figure 210: Pin access for domains other than the application domain

7.32.6 DPPI access control

Access to DPPI channels can be restricted. A channel can be declared as secure so that only secure peripherals can access it.

The security attribute of a DPPI channel is configured in DPPI[n].PERM (n=0..0) on page 603. When the secure attribute is set for a channel, only peripherals that have the secure attribute set will be able to publish events to this channel or subscribe to this channel to receive tasks.

The DPPI controller peripheral (DPPIC) is a split security peripheral, i.e., its security behavior depends on the security attributes of both the DPPIC and the accessing party. See Special considerations regarding the DPPIC configuration registers on page 594 for more information about the DPPIC security behavior.

If a non-secure peripheral wants to publish an event on a secure DPPI channel, the channel will ignore the event. If a non-secure peripheral subscribes to a secure DPPI channel, it will not receive any events from

this channel. The following figure illustrates the principle of operation of the security logic for a subscribed channel:

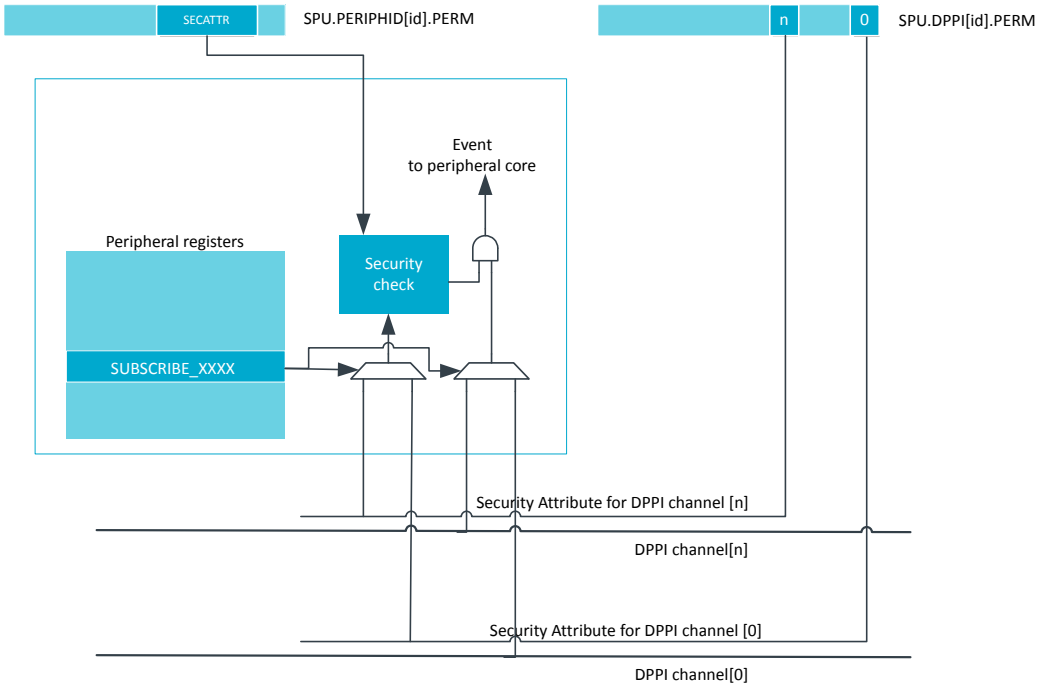

Figure 211: Subscribed channel security concept

No error reporting mechanism is associated with the DPPI access control logic.

7.32.6.1 Special considerations regarding the DPPIC configuration registers

DPPI channels can be enabled, disabled and grouped through the DPPI controller (DPPIC). The DPPIC is a split-security peripheral, and handles both secure and non-secure accesses.

A non-secure peripheral access will only be able to configure and control DPPI channels defined as non-secure in SPU's DPPI[n].PERM register(s). A secure peripheral access can control all DPPI channels, independently of the configuration in the DPPI[n].PERM register(s).

The DPPIC allows the creation of group of channels to be able to enable or disable all channels within a group simultaneously. The security attribute of a group of channels (secure or non-secure) is defined as follows:

- If all channels (enabled or not) in the group are non-secure, then the group is considered non-secure
- If at least one of the channels (enabled or not) in the group is secure, then the group is considered secure

A non-secure access to a DPPIC register, or a bitfield controlling a channel marked as secure in DPPI[n].PERM register(s), will be ignored:

- Write accesses will have no effect
- Read will always return a zero value

No exceptions are thrown when a non-secure access targets a register or bitfield controlling a secure channel. For example, if the bit *i* is set in the DPPI[n].PERM register (declaring the DPPI channel *i* as secure), then:

- Non-secure write accesses to registers CHEN, CHENSET and CHENCLR will not be able to write to bit *i* of those registers

- Non-secure write accesses to registers TASK_CHG[*j*].EN and TASK_CHG[*j*].DIS will be ignored if the channel group *j* contains at least one channel defined as secure (it can be the channel *i* itself or any channel declared as secured)
- Non-secure read accesses to registers CHEN, CHENSET and CHENCLR will always read zero for the bit at position *i*

For the channel configuration registers (DPPIC.CHG[n]), access from non-secure code is only possible if the included channels are all non-secure, whether the channels are enabled or not. If a DPPIC.CHG[*g*] register included one or more secure channels, then the group *g* is considered as secure and only a secure transfer can read or write DPPIC.CHG[*g*]. A non-secure write will be ignored and a non-secure read will return zero.

The DPPIC can subscribe to secure or non-secure channels through SUBSCRIBE_CHG[n] registers in order to trigger task for enabling or disabling groups of channels. But an event from a non-secure channel will be ignored if the group subscribing to this channel is secure. An event from a secure channel can trigger both secure and non-secure tasks.

7.32.7 External domain access control

Other domains with their own CPUs can access peripherals, flash and RAM memories. The SPU allows controlling accesses from those bus masters.

The external domains can access application MCU memories and peripherals. External domains are assigned security attributes as described in register EXTDOMAIN[n].PERM.

Domain	Capability register	Permission register
Network MCU	EXTDOMAIN[n].PERM (n=0..0) on page 602, SECUREMAPPING field	EXTDOMAIN[n].PERM (n=0..0) on page 602, SECATTR field

Table 156: Register mapping for external domains

The figure below illustrates how the security control units are used to assign security attributes to transfers initiated by the external domains:

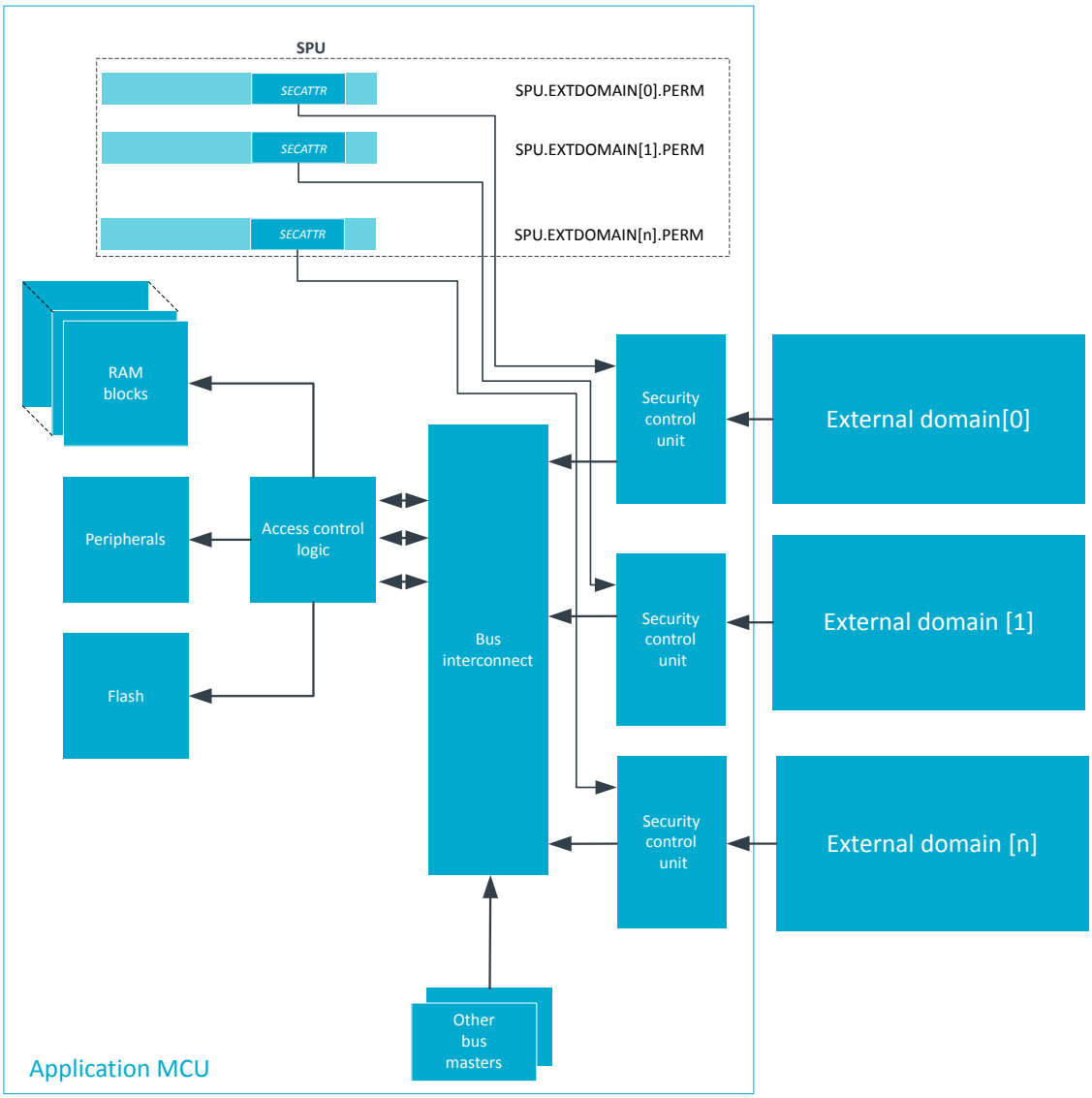

Figure 212: Access control from external domains

7.32.8 Arm TrustZone for Cortex-M ID allocation

Flash and RAM regions, as well as non-secure and secure peripherals, are assigned unique Arm TrustZone IDs.

The Arm TrustZone ID should not be mistaken for the peripheral ID used to identify peripherals.

The following table lists the Arm TrustZone ID allocation.

Regions	Arm TrustZone Cortex-M ID
Flash regions 0..63	0..63
RAM regions 0..63	64..127
UICR	252
FICR	252
CACHEDATA	252
CACHEINFO	252
Non-secure peripherals	253
Secure peripherals	254

Table 157: Arm TrustZone ID allocation

7.32.9 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50003000	APPLICATION	SPU	SPU	S	NA	System protection unit	

Table 158: Instances

Register	Offset	Security	Description	
EVENTS_RAMACCERR	0x100		A security violation has been detected for the RAM memory space	
EVENTS_FLASHACCERR	0x104		A security violation has been detected for the flash memory space	
EVENTS_PERIPHACCERR	0x108		A security violation has been detected on one or several peripherals	
PUBLISH_RAMACCERR	0x180		Publish configuration for event RAMACCERR	
PUBLISH_FLASHACCERR	0x184		Publish configuration for event FLASHACCERR	
PUBLISH_PERIPHACCERR	0x188		Publish configuration for event PERIPHACCERR	
INTEN	0x300		Enable or disable interrupt	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
CAP	0x400		Show implemented features for the current device	
CPULOCK	0x404		Configure bits to lock down CPU features at runtime	
EXTDOMAIN[n].PERM	0x440		Access for bus access generated from the external domain n List capabilities of the external domain n	
DPPI[n].PERM	0x480		Select between secure and non-secure attribute for the DPPI channels	
DPPI[n].LOCK	0x484		Prevent further modification of the corresponding PERM register	
GPIOPORT[n].PERM	0x4C0		Select between secure and non-secure attribute for pins 0 to 31 of port n	Retained
GPIOPORT[n].LOCK	0x4C4		Prevent further modification of the corresponding PERM register	
FLASHNSC[n].REGION	0x500		Define which flash region can contain the non-secure callable (NSC) region n	
FLASHNSC[n].SIZE	0x504		Define the size of the non-secure callable (NSC) region n	
RAMNSC[n].REGION	0x540		Define which RAM region can contain the non-secure callable (NSC) region n	
RAMNSC[n].SIZE	0x544		Define the size of the non-secure callable (NSC) region n	
FLASHREGION[n].PERM	0x600		Access permissions for flash region n	
RAMREGION[n].PERM	0x700		Access permissions for RAM region n	
PERIPHID[n].PERM	0x800		List capabilities and access permissions for the peripheral with ID n	

Table 159: Register overview

7.32.9.1 EVENTS_RAMACCERR

Address offset: 0x100

A security violation has been detected for the RAM memory space

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RAMACCERR			A security violation has been detected for the RAM memory space
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.32.9.2 EVENTS_FLASHACCERR

Address offset: 0x104

A security violation has been detected for the flash memory space

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_FLASHACCERR			A security violation has been detected for the flash memory space
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.32.9.3 EVENTS_PERIPHACCERR

Address offset: 0x108

A security violation has been detected on one or several peripherals

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_PERIPHACCERR			A security violation has been detected on one or several peripherals
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.32.9.4 PUBLISH_RAMACCERR

Address offset: 0x180

Publish configuration for event RAMACCERR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RAMACCERR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.32.9.5 PUBLISH_FLASHACCERR

Address offset: 0x184

Publish configuration for event FLASHACCERR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event FLASHACCERR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.32.9.6 PUBLISH_PERIPHACCERR

Address offset: 0x188

Publish configuration for event PERIPHACCERR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event PERIPHACCERR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.32.9.7 INTEN

Address offset: 0x300

Enable or disable interrupt

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

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

ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	PERIPHACCERR			Enable or disable interrupt for event PERIPHACCERR
			Disabled	0	Disable
			Enabled	1	Enable

7.32.9.8 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	RAMACCERR			Write '1' to enable interrupt for event RAMACCERR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FLASHACCERR			Write '1' to enable interrupt for event FLASHACCERR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	PERIPHACCERR			Write '1' to enable interrupt for event PERIPHACCERR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.32.9.9 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	RAMACCERR			Write '1' to disable interrupt for event RAMACCERR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	FLASHACCERR			Write '1' to disable interrupt for event FLASHACCERR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	PERIPHACCERR			Write '1' to disable interrupt for event PERIPHACCERR

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

ID	R/W	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.32.9.10 CAP

Address offset: 0x400

Show implemented features for the current device

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

ID	R/W	Field	Value ID	Value	Description
A	R	TZM			Show Arm TrustZone status
			NotAvailable	0	Arm TrustZone support not available
			Enabled	1	Arm TrustZone support is available

7.32.9.11 CPULOCK

Address offset: 0x404

Configure bits to lock down CPU features at runtime

Write '1' to any position to set the corresponding lock bit, which will remain set until the next reset

Any '0' writes to this register will be ignored

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LOCKSVTAIRCR			Write '1' to prevent updating the secure interrupt configuration until the next reset When set to '1', this lock bit prevents changes to: • The Secure vector table base address • Handling of Secure interrupt priority • BusFault, HardFault, and NMI security target
			Locked	1	Disables writes to the VTOR_S, AIRCR.PRIS, and AIRCR.BFHFNMINS registers
			Unlocked	0	These registers can be updated
B	RW	LOCKNSVTOR			Write '1' to prevent updating the non-secure vector table base address until the next reset When set to '1', this lock bit prevents changes to the Non-secure interrupt vector table base address register VTOR_NS
			Locked	1	The address of the non-secure vector table is locked
			Unlocked	0	The address of the non-secure vector table can be updated

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

ID	R/W	Field	Value ID	Value	Description
C	RW	LOCKSMPU			Write '1' to prevent updating the secure MPU regions until the next reset When set to '1', this lock bit prevents changes to programmed Secure MPU memory regions and all writes to the registers are ignored
			Locked	1	Disables writes to the MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An and MPU_RLAR_An from software or from a debug agent connected to the processor in Secure state
			Unlocked	0	These registers can be updated
D	RW	LOCKNSMPU			Write '1' to prevent updating the Non-secure MPU regions until the next reset When set to '1', this lock bit prevents changes to programmed Non-secure MPU memory regions already programmed. All writes to the registers are ignored.
			Locked	1	Disables writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn from software or from a debug agent connected to the processor
			Unlocked	0	These registers can be updated
E	RW	LOCKSAU			Write '1' to prevent updating the secure SAU regions until the next reset When set to '1', this lock bit prevents changes to Secure SAU memory regions already programmed. All writes to the registers are ignored.
			Locked	1	Disables writes to the SAU_CTRL, SAU_RNR, SAU_RBAR and SAU_RLAR registers from software or from a debug agent connected to the processor
			Unlocked	0	These registers can be updated

7.32.9.12 EXTDOMAIN[n].PERM (n=0..0)

Address offset: 0x440 + (n × 0x4)

Access for bus access generated from the external domain n

List capabilities of the external domain n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A
Reset 0x00000002				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0**

ID	R/W	Field	Value ID	Value	Description
A	R	SECUREMAPPING			Define configuration capabilities for TrustZone Cortex-M secure attribute
					This does not affect DPPI in the external domain
			NonSecure	0	The bus access from this external domain always have the non-secure attribute set
			Secure	1	The bus access from this external domain always have the secure attribute set
			UserSelectable	2	Non-secure or secure attribute for bus access from this domain is defined by the EXTDOMAIN[n].PERM register
B	RW	SECATTR			Peripheral security mapping
					This bit has effect only if EXTDOMAIN[n].PERM.SECUREMAPPING reads as UserSelectable
			NonSecure	0	Bus accesses from this domain have the non-secure attribute set
			Secure	1	Bus accesses from this domain have secure attribute set
C	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.13 DPPI[n].PERM (n=0..0)

Address offset: 0x480 + (n × 0x8)

Select between secure and non-secure attribute for the DPPI 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				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 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**

ID	R/W	Field	Value ID	Value	Description
A-f	RW	CHANNEL[i] (i=0..31)			Select secure attribute
			Secure	1	Channel i has its secure attribute set
			NonSecure	0	Channel i has its non-secure attribute set

7.32.9.14 DPPI[n].LOCK (n=0..0)

Address offset: 0x484 + (n × 0x8)

Prevent further modification of the corresponding PERM register

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LOCK			
			Locked	1	DPPI[n].PERM register can't be changed until next reset
			Unlocked	0	DPPI[n].PERM register content can be changed

7.32.9.15 GPIOPORT[n].PERM (n=0..1) (Retained)

Address offset: 0x4C0 + (n × 0x8)

This register is a retained register

Select between secure and non-secure attribute for pins 0 to 31 of port n

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

7.32.9.16 GPIOPORT[n].LOCK (n=0..1)

Address offset: 0x4C4 + (n × 0x8)

Prevent further modification of the corresponding PERM register

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																			A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value			Description																												
A	RW	LOCK																																	
			Locked	1			GPIOPORT[n].PERM register can't be changed until next reset																														
			Unlocked	0			GPIOPORT[n].PERM register content can be changed																														

7.32.9.17 FLASHNSC[n].REGION (n=0..1)

Address offset: 0x500 + (n × 0x8)

Define which flash region can contain the non-secure callable (NSC) region n

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																											B				A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value			Description																												
A	RW	REGION					Region number																												
B	RW	LOCK																																	
			Unlocked	0			This register can be updated																												
			Locked	1			The content of this register can't be changed until the next reset																												

7.32.9.18 FLASHNSC[n].SIZE (n=0..1)

Address offset: 0x504 + (n × 0x8)

Define the size of the non-secure callable (NSC) region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	SIZE			Size of the non-secure callable (NSC) region n
			Disabled	0	The region n is not defined as a non-secure callable region. Normal security attributes (secure or non-secure) are enforced.
			32	1	The region n is defined as non-secure callable with size 32 bytes
			64	2	The region n is defined as non-secure callable with size 64 bytes
			128	3	The region n is defined as non-secure callable with size 128 bytes
			256	4	The region n is defined as non-secure callable with size 256 bytes
			512	5	The region n is defined as non-secure callable with size 512 bytes
			1024	6	The region n is defined as non-secure callable with size 1024 bytes
			2048	7	The region n is defined as non-secure callable with size 2048 bytes
			4096	8	The region n is defined as non-secure callable with size 4096 bytes
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.19 RAMNSC[n].REGION (n=0..1)

Address offset: 0x540 + (n × 0x8)

Define which RAM region can contain the non-secure callable (NSC) region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	REGION			Region number
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.20 RAMNSC[n].SIZE (n=0..1)

Address offset: 0x544 + (n × 0x8)

Define the size of the non-secure callable (NSC) region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	SIZE			Size of the non-secure callable (NSC) region n
			Disabled	0	The region n is not defined as a non-secure callable region. Normal security attributes (secure or non-secure) are enforced.
			32	1	The region n is defined as non-secure callable with size 32 bytes
			64	2	The region n is defined as non-secure callable with size 64 bytes
			128	3	The region n is defined as non-secure callable with size 128 bytes
			256	4	The region n is defined as non-secure callable with size 256 bytes
			512	5	The region n is defined as non-secure callable with size 512 bytes
			1024	6	The region n is defined as non-secure callable with size 1024 bytes
			2048	7	The region n is defined as non-secure callable with size 2048 bytes
			4096	8	The region n is defined as non-secure callable with size 4096 bytes
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.21 FLASHREGION[n].PERM (n=0..63)

Address offset: 0x600 + (n × 0x4)

Access permissions for flash region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000017				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	EXECUTE			Configure instruction fetch permissions from flash region n
			Enable	1	Allow instruction fetches from flash region n
			Disable	0	Block instruction fetches from flash region n
B	RW	WRITE			Configure write permission for flash region n
			Enable	1	Allow write operation to region n
			Disable	0	Block write operation to region n
C	RW	READ			Configure read permissions for flash region n
			Enable	1	Allow read operation from flash region n
			Disable	0	Block read operation from flash region n
D	RW	SECATTR			Security attribute for flash region n
			Non_Secure	0	Flash region n security attribute is non-secure
			Secure	1	Flash region n security attribute is secure
E	RW	LOCK			
			Unlocked	0	This register can be updated

Bit number				31 30 29 28 27 26 25 24 23 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 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 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.22 RAMREGION[n].PERM (n=0..63)

Address offset: 0x700 + (n × 0x4)

Access permissions for RAM region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E 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 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	EXECUTE			Configure instruction fetch permissions from RAM region n
			Enable	1	Allow instruction fetches from RAM region n
			Disable	0	Block instruction fetches from RAM region n
B	RW	WRITE			Configure write permission for RAM region n
			Enable	1	Allow write operation to RAM region n
			Disable	0	Block write operation to RAM region n
C	RW	READ			Configure read permissions for RAM region n
			Enable	1	Allow read operation from RAM region n
			Disable	0	Block read operation from RAM region n
D	RW	SECATTR			Security attribute for RAM region n
			Non_Secure	0	RAM region n security attribute is non-secure
			Secure	1	RAM region n security attribute is secure
E	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

7.32.9.23 PERIPHID[n].PERM (n=0..255)

Address offset: 0x800 + (n × 0x4)

List capabilities and access permissions for the peripheral with ID n

Note: Reset values are unique per peripheral instantation. Please refer to the peripheral instantiation table. Entries not listed in the instantiation table are undefined.

Bit number				31 30 29 28 27 26 25 24 23 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 B A A
Reset 0x00000012				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	R	SECUREMAPPING			Define configuration capabilities for Arm TrustZone Cortex-M secure attribute
			NonSecure	0	This peripheral is always accessible as a non-secure peripheral
			Secure	1	This peripheral is always accessible as a secure peripheral
			UserSelectable	2	Non-secure or secure attribute for this peripheral is defined by the PERIPHID[n].PERM 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 B A A
Reset 0x00000012				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0**
ID	R/W	Field	Value ID	Value	Description
			Split	3	This peripheral implements the split security mechanism. Non-secure or secure attribute for this peripheral is defined by the PERIPHID[n].PERM register.
B	R	DMA			Indicates if the peripheral has DMA capabilities and if DMA transfer can be assigned to a different security attribute than the peripheral itself
			NoDMA	0	Peripheral has no DMA capability
			NoSeparateAttribute	1	Peripheral has DMA and DMA transfers always have the same security attribute as assigned to the peripheral
			SeparateAttribute	2	Peripheral has DMA and DMA transfers can have a different security attribute than the one assigned to the peripheral
C	RW	SECATTR			Peripheral security mapping This bit has effect only if PERIPHID[n].PERM.SECUREMAPPING reads as UserSelectable or Split
			Secure	1	Peripheral is mapped in secure peripheral address space
			NonSecure	0	If SECUREMAPPING == UserSelectable: Peripheral is mapped in non-secure peripheral address space. If SECUREMAPPING == Split: Peripheral is mapped in non-secure and secure peripheral address space.
D	RW	DMASEC			Security attribution for the DMA transfer This bit has effect only if PERIPHID[n].PERM.SECATTR is set to secure
			Secure	1	DMA transfers initiated by this peripheral have the secure attribute set
			NonSecure	0	DMA transfers initiated by this peripheral have the non-secure attribute set
E	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset
F	R	PRESENT			Indicate if a peripheral is present with ID n
			NotPresent	0	Peripheral is not present
			IsPresent	1	Peripheral is present

7.33 SWI — Software interrupts

A set of interrupts have been reserved for use as software interrupts.

7.33.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x4101A000	NETWORK	SWI	SWI0	NS	NA	Software interrupt 0	
0x4101B000	NETWORK	SWI	SWI1	NS	NA	Software interrupt 1	
0x4101C000	NETWORK	SWI	SWI2	NS	NA	Software interrupt 2	
0x4101D000	NETWORK	SWI	SWI3	NS	NA	Software interrupt 3	

Table 160: Instances

7.34 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.

The main features of TEMP are the following:

- Temperature range is greater than or equal to operating temperature of the device
- Resolution is 0.25 °C

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 69 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.

7.34.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x41010000	NETWORK	TEMP	TEMP	NS	NA	Temperature sensor	

Table 161: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start temperature measurement
TASKS_STOP	0x004		Stop temperature measurement
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
EVENTS_DATARDY	0x100		Temperature measurement complete, data ready
PUBLISH_DATARDY	0x180		Publish configuration for event DATARDY
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
TEMP	0x508		Temperature in °C (0.25° steps)
A0	0x520		Slope of first piecewise linear function
A1	0x524		Slope of second piecewise linear function
A2	0x528		Slope of third piecewise linear function
A3	0x52C		Slope of fourth piecewise linear function
A4	0x530		Slope of fifth piecewise linear function

Register	Offset	Security	Description
A5	0x534		Slope of sixth piecewise linear function
B0	0x540		y-intercept of first piecewise linear function
B1	0x544		y-intercept of second piecewise linear function
B2	0x548		y-intercept of third piecewise linear function
B3	0x54C		y-intercept of fourth piecewise linear function
B4	0x550		y-intercept of fifth piecewise linear function
B5	0x554		y-intercept of sixth piecewise linear function
T0	0x560		Endpoint of first piecewise linear function
T1	0x564		Endpoint of second piecewise linear function
T2	0x568		Endpoint of third piecewise linear function
T3	0x56C		Endpoint of fourth piecewise linear function
T4	0x570		Endpoint of fifth piecewise linear function

Table 162: Register overview

7.34.1.1 TASKS_START

Address offset: 0x000

Start temperature measurement

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start temperature measurement
			Trigger	1	Trigger task

7.34.1.2 TASKS_STOP

Address offset: 0x004

Stop temperature measurement

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop temperature measurement
			Trigger	1	Trigger task

7.34.1.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.34.1.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.34.1.5 EVENTS_DATARDY

Address offset: 0x100

Temperature measurement complete, data ready

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_DATARDY			Temperature measurement complete, data ready
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.34.1.6 PUBLISH_DATARDY

Address offset: 0x180

Publish configuration for event DATARDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event DATARDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.34.1.7 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	DATARDY			Write '1' to enable interrupt for event DATARDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.34.1.8 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	DATARDY			Write '1' to disable interrupt for event DATARDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.34.1.9 TEMP

Address offset: 0x508

Temperature in °C (0.25° steps)

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

ID	R/W	Field	Value ID	Value	Description
A	R	TEMP			Temperature in °C (0.25° steps) Result of temperature measurement. Die temperature in °C, 2's complement format, 0.25 °C steps. Decision point: DATARDY

7.34.1.10 A0

Address offset: 0x520

Slope of first piecewise 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
Reset 0x000002D9	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	A0			Slope of first piecewise linear function

7.34.1.11 A1

Address offset: 0x524

Slope of second piecewise 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
Reset 0x00000322	**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 1 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	A1			Slope of second piecewise linear function

7.34.1.12 A2

Address offset: 0x528

Slope of third piecewise 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
Reset 0x00000355	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	A2			Slope of third piecewise linear function

7.34.1.13 A3

Address offset: 0x52C

Slope of fourth piecewise 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
Reset 0x000003DF	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	A3			Slope of fourth piecewise linear function

7.34.1.14 A4

Address offset: 0x530

Slope of fifth piecewise 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
Reset 0x0000044E	**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 1 0 0 1 1 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	A4			Slope of fifth piecewise linear function

7.34.1.15 A5

Address offset: 0x534

Slope of sixth piecewise 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
Reset 0x000004B7					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	A5			Slope of sixth piecewise linear function

7.34.1.16 B0

Address offset: 0x540

y-intercept of first piecewise 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
Reset 0x00000FC7					**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 0 0 0 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	B0			y-intercept of first piecewise linear function

7.34.1.17 B1

Address offset: 0x544

y-intercept of second piecewise 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
Reset 0x00000F71					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	B1			y-intercept of second piecewise linear function

7.34.1.18 B2

Address offset: 0x548

y-intercept of third piecewise 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
Reset 0x00000F6C					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 1 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	B2			y-intercept of third piecewise linear function

7.34.1.19 B3

Address offset: 0x54C

y-intercept of fourth piecewise 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
Reset 0x00000FCB				**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 0 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	B3			y-intercept of fourth piecewise linear function

7.34.1.20 B4

Address offset: 0x550

y-intercept of fifth piecewise 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
Reset 0x0000004B				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	B4			y-intercept of fifth piecewise linear function

7.34.1.21 B5

Address offset: 0x554

y-intercept of sixth piecewise 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
Reset 0x000000F6				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	B5			y-intercept of sixth piecewise linear function

7.34.1.22 T0

Address offset: 0x560

Endpoint of first piecewise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A
Reset 0x000000E1				**0 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 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	T0			Endpoint of first piecewise linear function

7.34.1.23 T1

Address offset: 0x564

Endpoint of second piecewise linear function

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A
Reset 0x000000F9				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	T1			Endpoint of second piecewise linear function

7.34.1.24 T2

Address offset: 0x568

Endpoint of third piecewise linear function

Bit number						31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID						A A A A A A A A
Reset 0x00000010						**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description	
A	RW	T2			Endpoint of third piecewise linear function	

7.34.1.25 T3

Address offset: 0x56C

Endpoint of fourth piecewise linear function

Bit number						31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID						A A A A A A A A
Reset 0x00000026						**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0**
ID	R/W	Field	Value ID	Value	Description	
A	RW	T3			Endpoint of fourth piecewise linear function	

7.34.1.26 T4

Address offset: 0x570

Endpoint of fifth piecewise linear function

Bit number						31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID						A A A A A A A A
Reset 0x0000003F						**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description	
A	RW	T4			Endpoint of fifth piecewise linear function	

7.34.2 Electrical specification

7.34.2.1 Temperature Sensor Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{TEMP}	Time required for temperature measurement		36		µs
$T_{TEMP,RANGE}$	Temperature sensor range	-20		70	°C
$T_{TEMP,ACC}$	Temperature sensor accuracy	-5		5	°C
$T_{TEMP,RANGE,EXT}$	Temperature sensor range, extended temperature range	-40		105	°C
$T_{TEMP,ACC,EXT}$	Temperature sensor accuracy, extended temperature range	-7		7	°C
$T_{TEMP,RES}$	Temperature sensor resolution		0.25		°C
$T_{TEMP,STB}$	Sample to sample stability at constant device temperature			±0.25	°C
$T_{TEMP,OFFST}$	Sample offset at 25°C	-2.5		2.5	°C

7.35 TIMER — Timer/counter

This peripheral is a general purpose timer allowing time intervals to be defined by user input. It can operate in two modes: Timer mode and Counter mode.

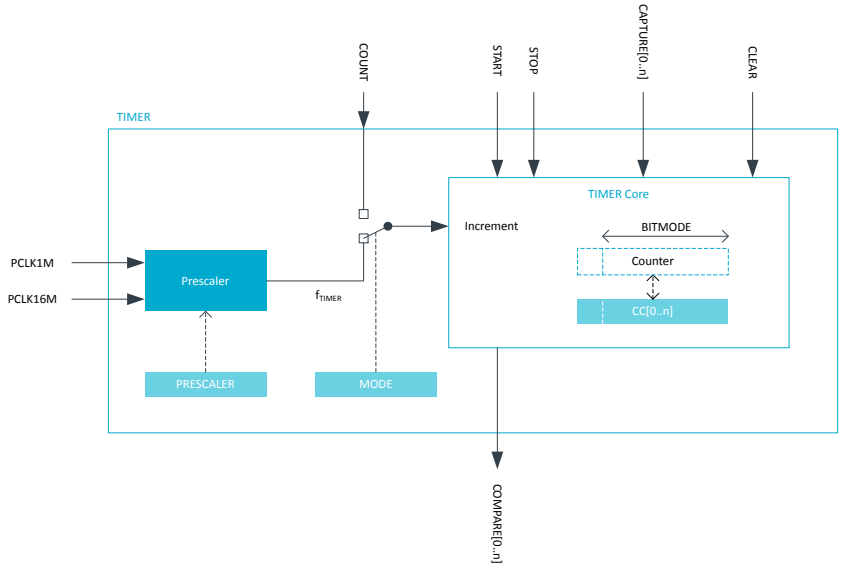

Figure 213: Block schematic for timer/counter

TIMER 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 the 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 on another system peripheral on the device. The PPI system also enables the TIMER task/event feature to generate periodic output and PWM signals to any GPIO. The number of GPIO inputs or outputs used at the same time is limited by the number of GPIOTE channels.

TIMER can operate in two modes: Timer mode and Counter mode. In both modes, TIMER is started by triggering the START task, and stopped by triggering the STOP task. After TIMER stops, it can resume timing/counting by triggering the START task again. When timing/counting resumes, TIMER continues from the value it was on prior to stopping.

In Timer mode, TIMER's internal Counter register is incremented by one for every tick of the timer frequency f_{TIMER}, as illustrated in Block schematic for timer/counter on page 617. The timer frequency is derived from PCLK16M as shown in the following example, using the values specified in the PRESCALER register.

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

When $f_{TIMER} \leq 1$ MHz, TIMER uses 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, meaning the timer frequency and the prescaler are not utilized in Counter mode. Similarly, the COUNT task has no effect in Timer mode.

TIMER's maximum value is configured by changing the bit-width of the timer in register BITMODE on page 625.

PRESCALER on page 625 and BITMODE on page 625 must only be updated when TIMER is stopped. If these registers are updated while TIMER is started, unpredictable behavior may occur.

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

The Counter register can be cleared by triggering the CLEAR task. This will explicitly set the internal value to zero.

TIMER implements multiple capture/compare registers.

Independent of prescaler settings, the accuracy of TIMER is equivalent to one tick of the timer frequency f_{TIMER} as illustrated in Block schematic for timer/counter on page 617.

7.35.1 Capture

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.

7.35.2 Compare

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

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 625 specifies how many Counter and capture/compare register bits are used when the comparison is performed. Other bits are ignored.

The COMPARE event can be configured to operate in one-shot mode by configuring the corresponding ONESHOTEN[n] register. After writing CC[n], a COMPARE[n] event is generated the first time the Counter matches CC[n].

7.35.3 Task delays

After TIMER is started, the CLEAR, COUNT, and STOP tasks are guaranteed to take effect within one clock cycle of the PCLK16M.

7.35.4 Task priority

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

If one or more of the CAPTURE tasks and the CLEAR task are triggered at the same time, that is, within the same period of PCLK16M, the CAPTURE tasks are prioritized. This means that the CC registers will capture the counter value before the CLEAR tasks are triggered.

7.35.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5000F000 0x4000F000	APPLICATION	TIMER	TIMER0 : S TIMER0 : NS	US	NA	Timer 0	6 capture compare channels implemented
0x50010000 0x40010000	APPLICATION	TIMER	TIMER1 : S TIMER1 : NS	US	NA	Timer 1	6 capture compare channels implemented
0x50011000 0x40011000	APPLICATION	TIMER	TIMER2 : S TIMER2 : NS	US	NA	Timer 2	6 capture compare channels implemented
0x4100C000	NETWORK	TIMER	TIMER0	NS	NA	Timer 0	
0x41018000	NETWORK	TIMER	TIMER1	NS	NA	Timer 1	
0x41019000	NETWORK	TIMER	TIMER2	NS	NA	Timer 2	

Table 163: Instances

Register	Offset	Security	Description	
TASKS_START	0x000		Start Timer	
TASKS_STOP	0x004		Stop Timer	
TASKS_COUNT	0x008		Increment Timer (Counter mode only)	
TASKS_CLEAR	0x00C		Clear time	
TASKS_SHUTDOWN	0x010		Shut down timer	Deprecated
TASKS_CAPTURE[n]	0x040		Capture Timer value to CC[n] register	
SUBSCRIBE_START	0x080		Subscribe configuration for task START	
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP	
SUBSCRIBE_COUNT	0x088		Subscribe configuration for task COUNT	
SUBSCRIBE_CLEAR	0x08C		Subscribe configuration for task CLEAR	
SUBSCRIBE_SHUTDOWN	0x090		Subscribe configuration for task SHUTDOWN	Deprecated
SUBSCRIBE_CAPTURE[n]	0x0C0		Subscribe configuration for task CAPTURE[n]	
EVENTS_COMPARE[n]	0x140		Compare event on CC[n] match	
PUBLISH_COMPARE[n]	0x1C0		Publish configuration for event COMPARE[n]	
SHORTS	0x200		Shortcuts between local events and tasks	
INTEN	0x300		Enable or disable interrupt	
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[n]	0x540		Capture/Compare register n	
ONESHOTEN[n]	0x580		Enable one-shot operation for Capture/Compare channel n	

Table 164: Register overview

7.35.5.1 TASKS_START

Address offset: 0x000

Start 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start Timer
			Trigger	1	Trigger task

7.35.5.2 TASKS_STOP

Address offset: 0x004

Stop 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop Timer
			Trigger	1	Trigger task

7.35.5.3 TASKS_COUNT

Address offset: 0x008

Increment Timer (Counter mode only)

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_COUNT			Increment Timer (Counter mode only)
			Trigger	1	Trigger task

7.35.5.4 TASKS_CLEAR

Address offset: 0x00C

Clear time

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_CLEAR			Clear time
			Trigger	1	Trigger task

7.35.5.5 TASKS_SHUTDOWN (Deprecated)

Address offset: 0x010

Shut down 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_SHUTDOWN			Shut down timer Deprecated
			Trigger	1	Trigger task

7.35.5.6 TASKS_CAPTURE[n] (n=0..7)

Address offset: 0x040 + (n × 0x4)

Capture Timer value to CC[n] register

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_CAPTURE			Capture Timer value to CC[n] register
			Trigger	1	Trigger task

7.35.5.7 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.8 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.9 SUBSCRIBE_COUNT

Address offset: 0x088

Subscribe configuration for task COUNT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task COUNT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.10 SUBSCRIBE_CLEAR

Address offset: 0x08C

Subscribe configuration for task 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				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CLEAR will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.11 SUBSCRIBE_SHUTDOWN (Deprecated)

Address offset: 0x090

Subscribe configuration for task SHUTDOWN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SHUTDOWN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.12 SUBSCRIBE_CAPTURE[n] (n=0..7)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task CAPTURE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task CAPTURE[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.35.5.13 EVENTS_COMPARE[n] (n=0..7)

Address offset: 0x140 + (n × 0x4)

Compare event on CC[n] match

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_COMPARE			Compare event on CC[n] match
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.35.5.14 PUBLISH_COMPARE[n] (n=0..7)

Address offset: 0x1C0 + (n × 0x4)

Publish configuration for event COMPARE[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event COMPARE[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.35.5.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

ID	R/W	Field	Value ID	Value	Description
A-H	RW	COMPARE[i]_CLEAR (i=0..7)			Shortcut between event COMPARE[i] and task CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
I-P	RW	COMPARE[i]_STOP (i=0..7)			Shortcut between event COMPARE[i] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.35.5.16 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A-H	RW	COMPARE[i] (i=0..7)			Enable or disable interrupt for event COMPARE[i]
			Disabled	0	Disable
			Enabled	1	Enable

7.35.5.17 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 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A-H	RW	COMPARE[i] (i=0..7)			Write '1' to enable interrupt for event COMPARE[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.35.5.18 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A-H	RW	COMPARE[i] (i=0..7)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.35.5.19 MODE

Address offset: 0x504

Timer mode selection

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

7.35.5.20 BITMODE

Address offset: 0x508

Configure the number of bits used by the TIMER

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

ID	R/W	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

7.35.5.21 PRESCALER

Address offset: 0x510

Timer prescaler register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	PRESCALER		[0..9]	Prescaler value

7.35.5.22 CC[n] (n=0..7)

Address offset: 0x540 + (n × 0x4)

Capture/Compare register n

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

ID	R/W	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.

7.35.5.23 ONESHOTEN[n] (n=0..7)

Address offset: 0x580 + (n × 0x4)

Enable one-shot operation for Capture/Compare channel n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ONESHOTEN			Enable one-shot operation
					Configures the corresponding compare-channel for one-shot operation
			Disable	0	Disable one-shot operation
					Compare event is generated every time the Counter matches CC[n]
			Enable	1	Enable one-shot operation
					Compare event is generated the first time the Counter matches CC[n] after CC[n] has been written

7.35.6 Electrical specification

7.36 TWIM — I²C compatible two-wire interface master with EasyDMA

TWI master with EasyDMA (TWIM) is a two-wire half-duplex master which can communicate with multiple slave devices connected to the same bus.

Listed here are the main features for TWIM:

- I²C compatible
- Supported baud rates: 100, 250, 400 and 1000 kbps
- Support for clock stretching (non I²C compliant)
- EasyDMA

The two-wire interface can communicate with a bi-directional wired-AND bus with two lines (SCL, SDA). The protocol makes it possible to interconnect up to 127 individually addressable devices. TWIM is not compatible with CBUS.

The GPIOs used for each two-wire interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

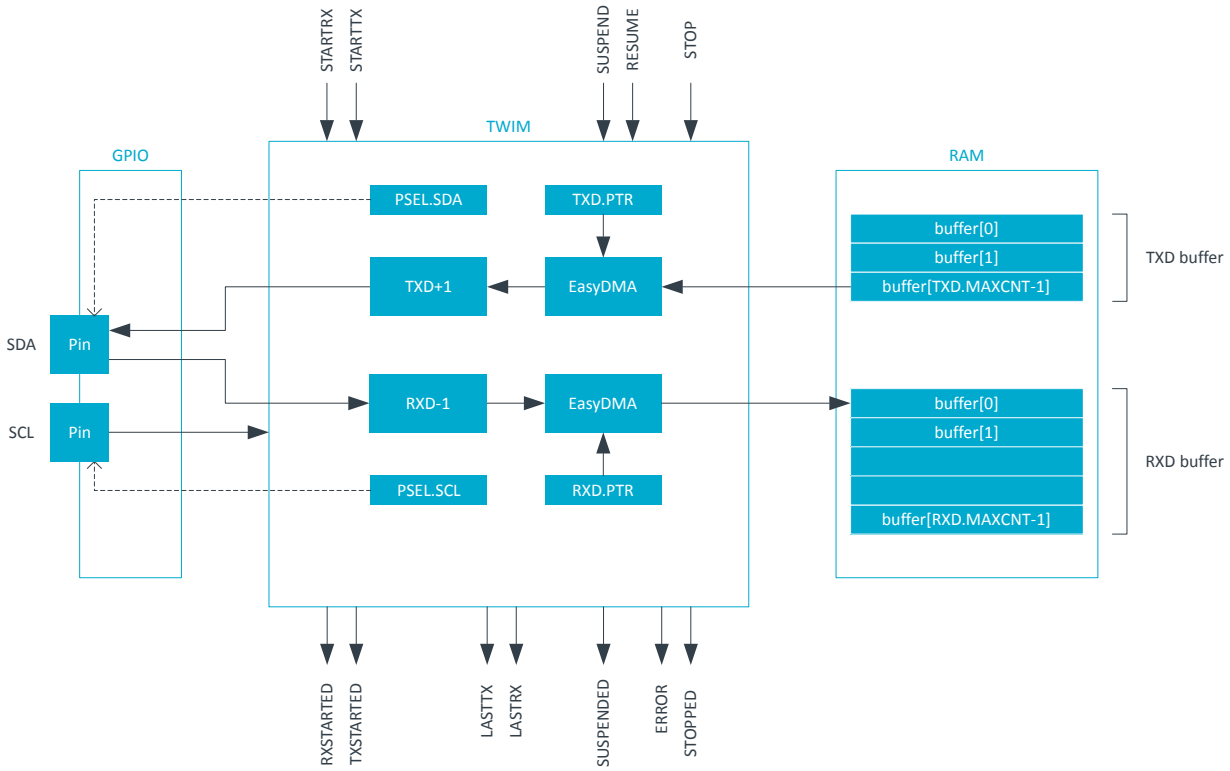

Figure 214: TWI master with EasyDMA

A typical TWI setup consists of one master and one or more slaves, as illustrated in the following figure. TWIM is only able to operate as a single master on the TWI bus. Multi-master bus configuration is not supported.

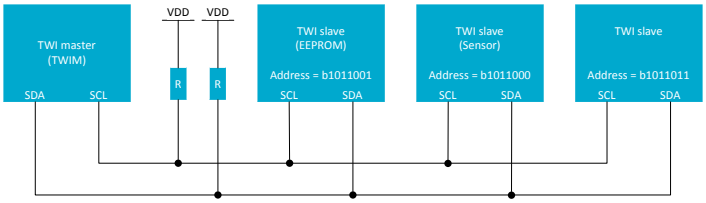

Figure 215: A typical TWI setup comprising one master and three slaves

TWIM supports clock stretching performed by the slaves. The SCK pulse following a stretched clock cycle may be shorter than specified by the I^2C specification.

TWIM is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task. After a STOP task, TWIM generates a STOPPED event when it has stopped.

After TWIM has been started, the STARTTX or STARTRX tasks should not be triggered again until TWIM has issued a LASTRX, LASTTX, or STOPPED event.

TWIM can be suspended using the SUSPEND task, such as when using the TWI master in a low priority interrupt context. When TWIM enters suspend state, it will automatically issue a SUSPENDED event while performing a continuous clock stretching until it is instructed to resume operation via a RESUME task. TWIM cannot be stopped while it is suspended, thus the STOP task has to be issued after the TWI master has been resumed.

Note: Any ongoing byte transfer will be allowed to complete before the suspend is enforced. A SUSPEND task has no effect unless TWIM is actively involved in a transfer.

If a NACK is clocked in from the slave, TWIM generates an ERROR event.

7.36.1 Shared resources

TWIM 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 TWIM before it can be configured and used.

Disabling a peripheral that has the same ID as TWIM will not reset any of the registers that are shared with TWIM. It is therefore important to configure all relevant registers explicitly to secure that TWIM operates correctly.

The Instantiation table in Peripherals on page 145 shows which peripherals have the same ID as the TWI.

7.36.2 EasyDMA

TWIM implements EasyDMA for accessing RAM without CPU involvement.

TWIM implements the EasyDMA channels found in the following table.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 165: TWIM EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 149.

The RXD.PTR, TXD.PTR, RXD.MAXCNT, and TXD.MAXCNT registers are double-buffered. They can be updated and prepared for the next RX or TX transmission immediately after having received the RXSTARTED or TXSTARTED event.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

7.36.3 Master write sequence

A TWIM write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, TWIM 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 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, TWIM clocks 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 TWIM will be followed by an ACK/NACK bit clocked in from the slave.

A typical TWIM write sequence is shown in the following figure, including clock stretching performed by TWIM following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect. This event can be used to synchronize the software.

TWIM will generate a LASTTX event when it starts to transmit the last byte.

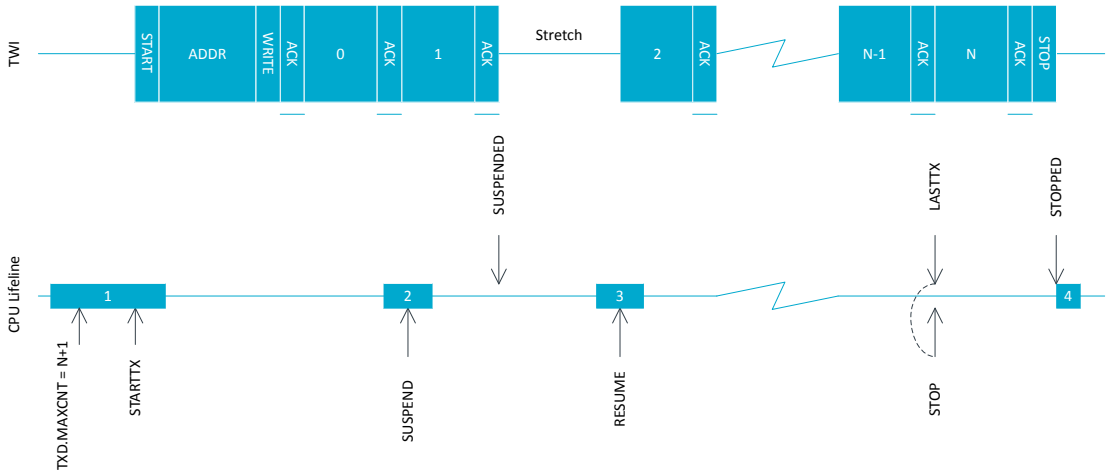

Figure 216: TWIM writing data to a slave

TWIM is stopped by triggering the STOP task. This task should be triggered during the transmission of the last byte to secure that TWIM will stop as fast as possible after sending the last byte. The shortcut between LASTTX and STOP can alternatively be used to accomplish this.

Note: TWIM does not stop by itself when the entire 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.

7.36.4 Master read sequence

A TWIM read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered, TWIM 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 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 sending the ACK bit, the TWI slave sends data to the master using the clock generated by TWIM.

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 have been received from the slave. TWIM generates a NACK after the last byte received to indicate that the read sequence shall stop.

A typical TWIM read sequence is illustrated in the following figure, including clock stretching performed by TWIM following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect. This event can be used to synchronize the software.

TWIM generates a LASTRX event when it is ready to receive the last byte. 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.

TWIM 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 recommended to use the shortcut between LASTRX and STOP to accomplish this.

TWIM 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.

TWIM cannot be stopped while suspended, so the STOP task must be issued after TWIM has been resumed.

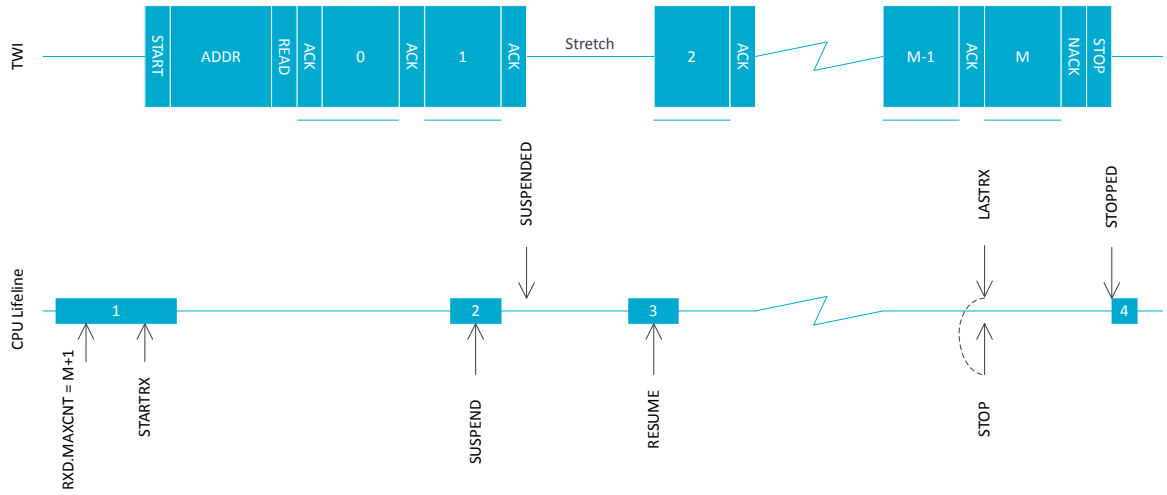

Figure 217: TWIM reading data from a slave

7.36.5 Master repeated start sequence

A typical repeated start sequence is when TWIM 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 following figure shows an example of a repeated start sequence where TWIM writes two bytes followed by reading four bytes from the slave.

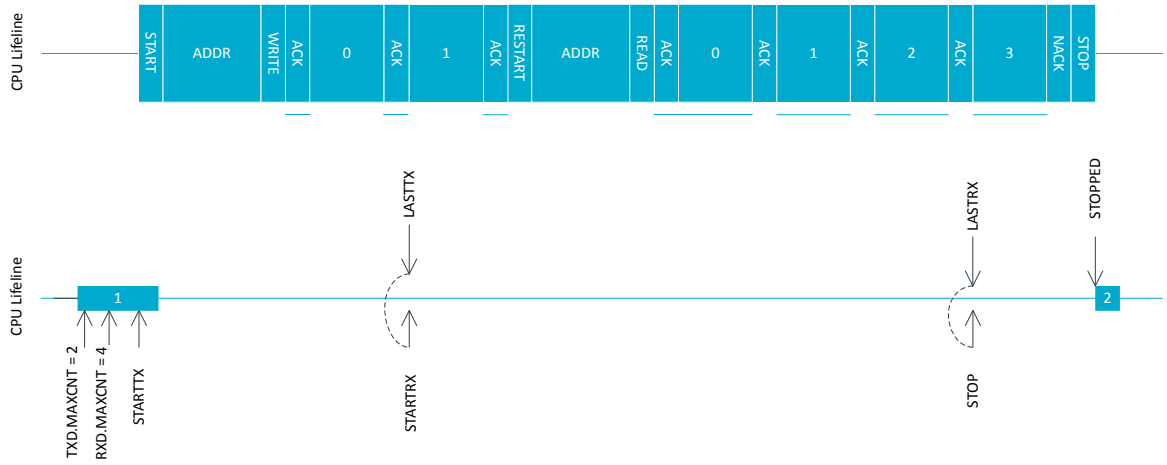

Figure 218: Master repeated start sequence

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. A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts is shown in the following figure.

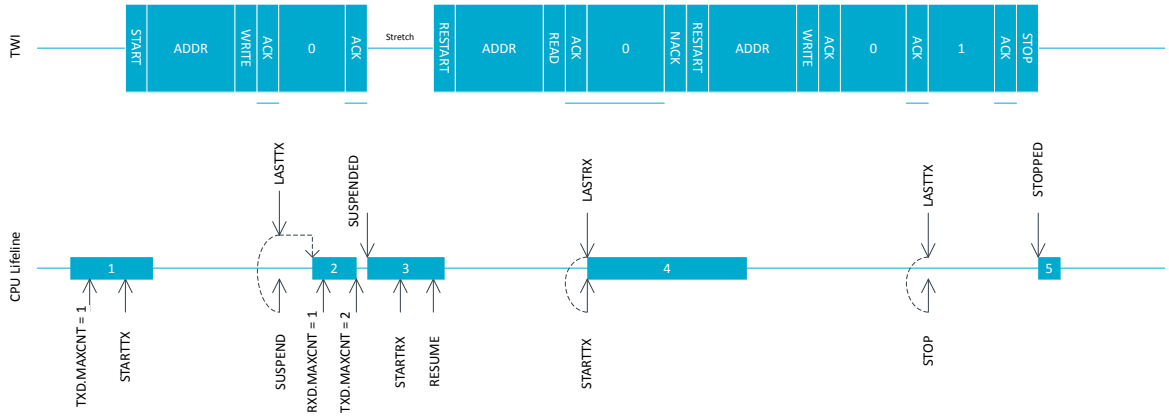

Figure 219: Double repeated start sequence

7.36.6 Low power

To ensure lowest possible power consumption when the peripheral is not needed stop and disable TWIM.

When the STOP task is sent, the software shall wait until the STOPPED event is received as a response before disabling the peripheral through the ENABLE register. If the peripheral is already stopped, the STOP task is not required.

7.36.7 Master mode pin configuration

The SCL and SDA signals are mapped to physical pins using the PSEL.SCL and PSEL.SDA registers.

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 System 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 master is disabled.

To secure correct signal levels on the pins used by TWIM while in System OFF mode, and when TWIM is disabled, these pins must be configured in the GPIO peripheral as described in the following table.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI master signal	TWI master pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL[27]	Input	Not applicable	S0D1[27]
SDA	As specified in PSEL.SDA[27]	Input	Not applicable	S0D1[27]

Table 166: GPIO configuration before enabling peripheral

[27] Special pin and drive strength considerations applies when using the 1000 kbps baud rate. For pin recommendations, see Pin assignments on page 783.

7.36.8 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	APPLICATION	TWIM	TWIM0 : S TWIM0 : NS	US	SA	Two-wire interface master 0	
0x50009000 0x40009000	APPLICATION	TWIM	TWIM1 : S TWIM1 : NS	US	SA	Two-wire interface master 1	
0x5000B000 0x4000B000	APPLICATION	TWIM	TWIM2 : S TWIM2 : NS	US	SA	Two-wire interface master 2	
0x5000C000 0x4000C000	APPLICATION	TWIM	TWIM3 : S TWIM3 : NS	US	SA	Two-wire interface master 3	
0x41013000	NETWORK	TWIM	TWIM0	NS	NA	Two-wire interface master 0	

Table 167: Instances

Register	Offset	Security	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
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
EVENTS_STOPPED	0x104		TWI stopped
EVENTS_ERROR	0x124		TWI error
EVENTS_SUSPENDED	0x148		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
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_SUSPENDED	0x1C8		Publish configuration for event SUSPENDED
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_LASTRX	0x1DC		Publish configuration for event LASTRX
PUBLISH_LASTTX	0x1E0		Publish configuration for event LASTTX
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4C4		Error source
ENABLE	0x500		Enable TWIM
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal
FREQUENCY	0x524		TWI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction

Register	Offset	Security	Description
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

Table 168: Register overview

7.36.8.1 TASKS_STARTRX

Address offset: 0x000

Start TWI receive sequence

Bit number																																
ID																																A
Reset 0x00000000																																**0**

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTRX			Start TWI receive sequence
			Trigger	1	Trigger task

7.36.8.2 TASKS_STARTTX

Address offset: 0x008

Start TWI transmit sequence

Bit number																																
ID																																A
Reset 0x00000000																																**0**

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start TWI transmit sequence
			Trigger	1	Trigger task

7.36.8.3 TASKS_STOP

Address offset: 0x014

Stop TWI transaction. Must be issued while the TWI master is not suspended.

Bit number																																
ID																																A
Reset 0x00000000																																**0**

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop TWI transaction. Must be issued while the TWI master is not suspended.
			Trigger	1	Trigger task

7.36.8.4 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

7.36.8.5 TASKS_RESUME

Address offset: 0x020

Resume TWI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

7.36.8.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.36.8.7 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.36.8.8 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.36.8.9 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.36.8.10 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.36.8.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.13 EVENTS_SUSPENDED

Address offset: 0x148

SUSPEND task has been issued, TWI traffic is now suspended.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_SUSPENDED			SUSPEND task has been issued, TWI traffic is now suspended.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.14 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.15 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.16 EVENTS_LASTRX

Address offset: 0x15C

Byte boundary, starting to receive the last byte

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_LASTRX			Byte boundary, starting to receive the last byte
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.17 EVENTS_LASTTX

Address offset: 0x160

Byte boundary, starting to transmit the last byte

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_LASTTX			Byte boundary, starting to transmit the last byte
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.36.8.18 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.19 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.20 PUBLISH_SUSPENDED

Address offset: 0x1C8

Publish configuration for event SUSPENDED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SUSPENDED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.21 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.22 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.23 PUBLISH_LASTRX

Address offset: 0x1DC

Publish configuration for event LASTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event LASTRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.24 PUBLISH_LASTTX

Address offset: 0x1E0

Publish configuration for event LASTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event LASTTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.36.8.25 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 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 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LASTTX_STARTRX			Shortcut between event LASTTX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	LASTTX_SUSPEND			Shortcut between event LASTTX and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LASTTX_STOP			Shortcut between event LASTTX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LASTRX_STARTTX			Shortcut between event LASTRX and task STARTTX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	LASTRX_STOP			Shortcut between event LASTRX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.36.8.26 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	SUSPENDED			Enable or disable interrupt for event SUSPENDED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	LASTRX			Enable or disable interrupt for event LASTRX
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	LASTTX			Enable or disable interrupt for event LASTTX
			Disabled	0	Disable
			Enabled	1	Enable

7.36.8.27 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to enable interrupt for event SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to enable interrupt for event LASTRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to enable interrupt for event LASTTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.36.8.28 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to disable interrupt for event SUSPENDED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to disable interrupt for event LASTRX
			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				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
J	RW	LASTTX			Write '1' to disable interrupt for event LASTTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.36.8.29 ERRORSRC

Address offset: 0x4C4

Error source

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

7.36.8.30 ENABLE

Address offset: 0x500

Enable TWIM

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

7.36.8.31 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

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

7.36.8.32 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

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

7.36.8.33 FREQUENCY

Address offset: 0x524

TWI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	FREQUENCY			TWI master clock frequency
			K100	0x01980000	100 kbps
			K250	0x04000000	250 kbps
			K400	0x06400000	400 kbps
			K1000	0x0FF00000	1000 kbps

7.36.8.34 RXD.PTR

Address offset: 0x534

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.36.8.35 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

7.36.8.36 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

7.36.8.37 RXD.LIST

Address offset: 0x540

EasyDMA list type

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.36.8.38 TXD.PTR

Address offset: 0x544

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.36.8.39 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

7.36.8.40 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

7.36.8.41 TXD.LIST

Address offset: 0x550

EasyDMA list type

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.36.8.42 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

7.36.9 Electrical specification

7.36.9.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL}$	Bit rates for TWIM[28]	100		1000	kbps
$t_{TWIM,START}$	Time from STARTRX/STARTTX task to transmission started		1.5		µs

7.36.9.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWIM,SU_DAT}	Data setup time before positive edge on SCL – all modes	20			ns
t_{TWIM,HD_DAT}	Data hold time after negative edge on SCL – 100, 250 and 400 kbps	500		625	ns
t_{TWIM,HD_DAT}	Data hold time after negative edge on SCL – 1000 kbps	250		315	ns
$t_{TWIM,HD\_STA,100kbps}$	TWIM master hold time for START and repeated START condition, 100 kbps	9900			ns
$t_{TWIM,HD\_STA,250kbps}$	TWIM master hold time for START and repeated START condition, 250 kbps	3900			ns
$t_{TWIM,HD\_STA,400kbps}$	TWIM master hold time for START and repeated START condition, 400 kbps	2400			ns
$t_{TWIM,HD\_STA,1000kbps}$	TWIM master hold time for START and repeated START condition, 1000 kbps	900			ns
$t_{TWIM,SU\_STO,100kbps}$	TWIM master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWIM,SU\_STO,250kbps}$	TWIM master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWIM,SU\_STO,400kbps}$	TWIM master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWIM,SU\_STO,1000kbps}$	TWIM master setup time from SCL high to STOP condition, 1000 kbps	500			ns
$t_{TWIM,BUF,100kbps}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5250			ns
$t_{TWIM,BUF,250kbps}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2250			ns
$t_{TWIM,BUF,400kbps}$	TWIM master bus free time between STOP and START conditions, 400 kbps	1500			ns
$t_{TWIM,BUF,1000kbps}$	TWIM master bus free time between STOP and START conditions, 1000 kbps	750			ns

[28] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO — General purpose input/output on page 219 for more details.

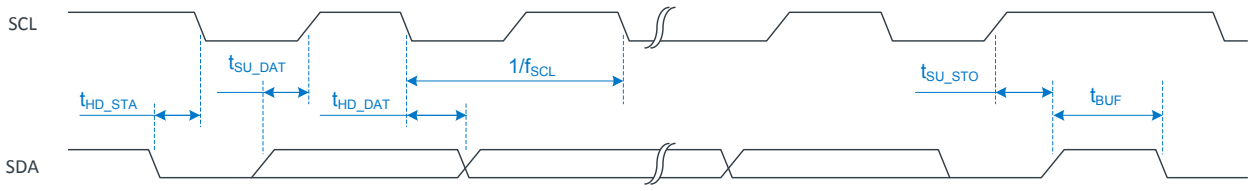

Figure 220: TWIM timing diagram, 1 byte transaction

7.36.10 Pullup resistor

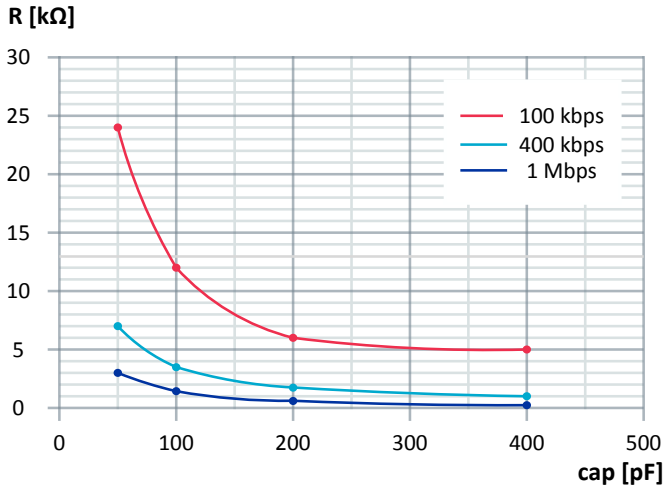

Figure 221: Recommended TWIM pullup value vs. line capacitance

- The I^2C specification allows a line capacitance of 400 pF at most.
- The value of internal pullup resistor (R_{PU}) for nRF5340 can be found in GPIO — General purpose input/output on page 219.

7.37 TWIS — I^2C compatible two-wire interface slave with EasyDMA

TWI slave with EasyDMA (TWIS) is a two-wire half-duplex slave which can communicate with a master device connected to the same bus.

Listed here are the main features for TWIS:

- I^2C compatible
- Supported baud rates: 100 and 400 kbps
- EasyDMA

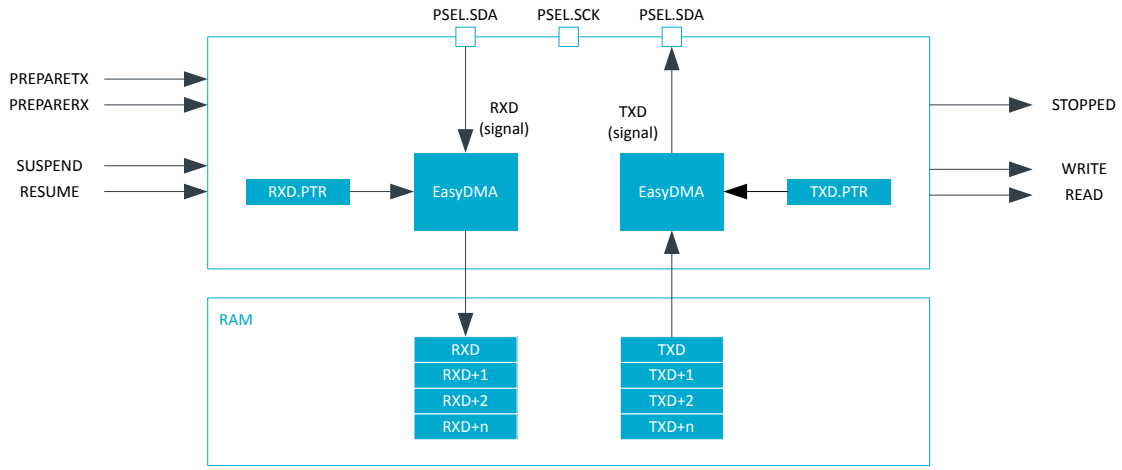

Figure 222: TWI slave with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see the following figure. TWIS is only able to operate with a single master on the TWI bus.

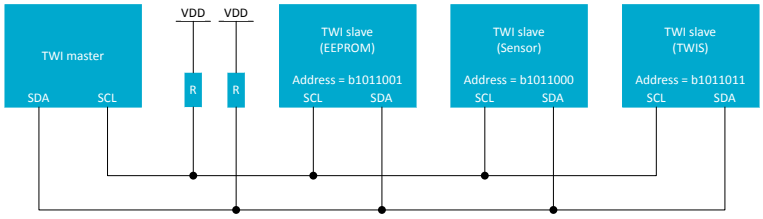

Figure 223: A typical TWI setup comprising one master and three slaves

The following figure shows the TWI slave state machine.

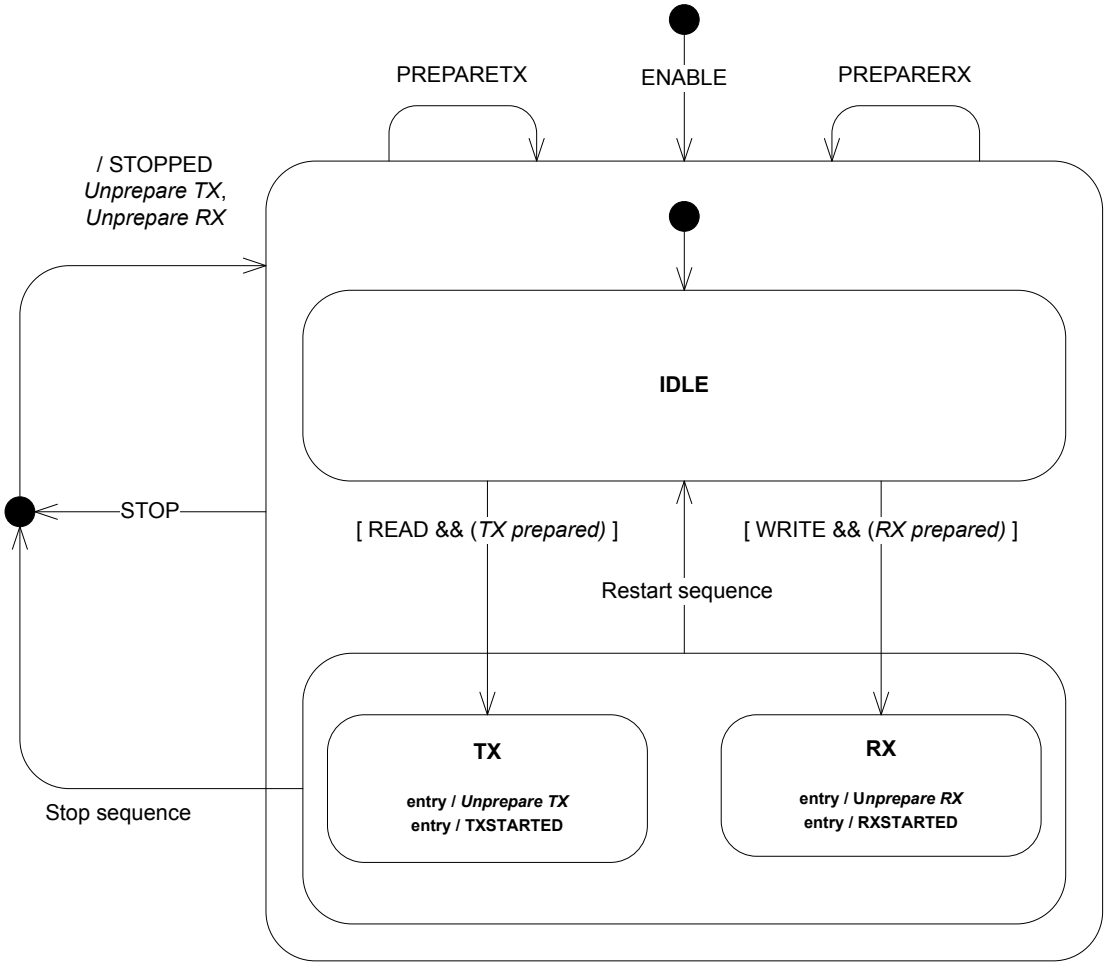

Figure 224: TWI slave state machine

The following table contains descriptions of the symbols used in the state machine.

Symbol	Type	Description
ENABLE	Register	The TWI slave has been enabled via the ENABLE register.
PREPARETX	Task	The TASKS_PREPARETX task has been triggered.
STOP	Task	The TASKS_STOP task has been triggered.
PREPARERX	Task	The TASKS_PREPARERX task has been triggered.
STOPPED	Event	The EVENTS_STOPPED event was generated.
RXSTARTED	Event	The EVENTS_RXSTARTED event was generated.
TXSTARTED	Event	The EVENTS_TXSTARTED event was generated.
TX prepared	Internal	Internal flag indicating that a TASKS_PREPARETX task has been triggered. This flag is not visible to the user.
RX prepared	Internal	Internal flag indicating that a TASKS_PREPARERX task has been triggered. This flag is not visible to the user.
Unprepare TX	Internal	Clears the internal 'TX prepared' flag until next TASKS_PREPARETX task.
Unprepare RX	Internal	Clears the internal 'RX prepared' flag until next TASKS_PREPARERX task.
Stop condition	TWI protocol	A TWI stop condition was detected.
Restart condition	TWI protocol	A TWI restart condition was detected.

Table 169: TWI slave state machine symbols

TWIS can perform clock stretching, with the premise that the master is able to support it.

It operates in a low power mode while waiting for a TWI master to initiate a transfer. As long as TWIS is not addressed, it will remain in this low power mode.

To secure correct behavior 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 behavior.

7.37.1 Shared resources

TWIS shares registers and other resources with other peripherals that have the same ID as TWIS.

Therefore, you must disable all peripherals that have the same ID as the TWI slave before TWIS 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 TWIS. It is therefore important to configure all relevant registers explicitly to secure that TWIS operates correctly.

The Instantiation table in Peripherals on page 145 shows which peripherals have the same ID as TWIS.

7.37.2 EasyDMA

TWIS implements EasyDMA for accessing RAM without CPU involvement.

The following table shows the Easy DMA channels that TWIS implements.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 170: TWIS EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 149.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

7.37.3 TWIS responding to a read command

Before TWIS can respond to a read command, it must be configured correctly and enabled via the ENABLE register. When enabled, TWIS will be in its IDLE state.

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.

TWIS is able to listen for up to two addresses at the same time. This is configured in the ADDRESS registers and the CONFIG register.

TWIS 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. TWIS will generate a READ event when it acknowledges the read command.

TWIS is only able to detect a read command from the IDLE state.

TWIS will set an internal 'TX prepared' flag when the PREPARETX task is triggered.

When the read command is received, TWIS 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, TWIS will stretch the master's clock until the PREPARETX task is triggered and the internal 'TX prepared' flag is set.

TWIS will generate the TXSTARTED event and clear the 'TX prepared' flag ('unprepare TX') when it enters the TX state. In this state TWIS will send the data bytes found in the transmit buffer to the master using the master's clock.

TWIS will go back to the IDLE state if the TWI slave receives a restart command when it is in the TX state.

TWIS is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. TWIS 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. TWIS 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 RXD.PTR, TXD.PTR, RXD.AMOUNT, and TXD.AMOUNT, are latched when the TXSTARTED event is generated.

TWIS can be forced to stop by triggering the STOP task. A STOPPED event will be generated when TWIS has stopped. TWIS 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 653.

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 TWIS read command response is illustrated in the following figure, including clock stretching following a SUSPEND task.

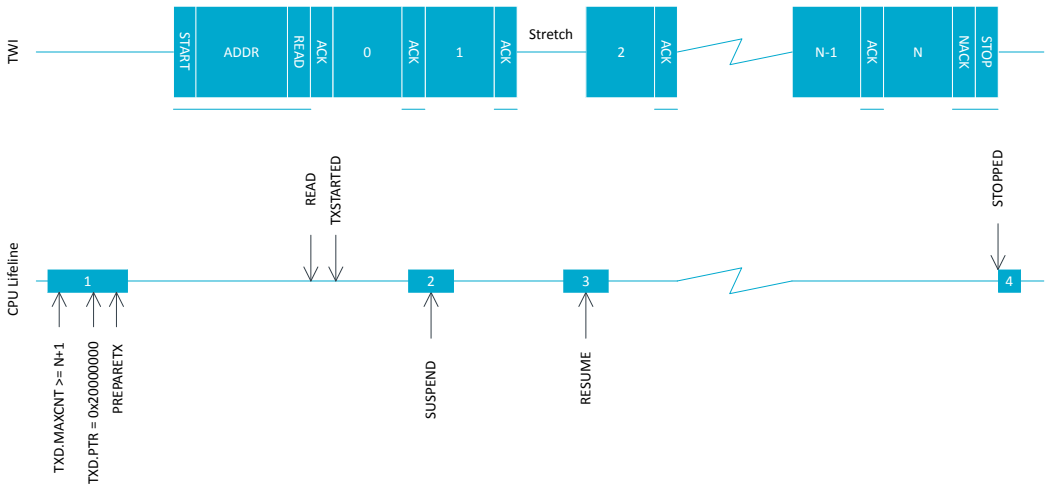

Figure 225: TWIS responding to a read command

7.37.4 TWIS responding to a write command

Before TWIS can respond to a write command, TWIS must be configured correctly and enabled via the ENABLE register. When enabled, TWIS will be in its IDLE state.

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.

TWIS is able to listen for up to two addresses at the same time. This is configured in the ADDRESS registers and the CONFIG register.

TWIS 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. TWIS will generate a WRITE event if it acknowledges the write command.

TWIS is only able to detect a write command from the IDLE state.

TWIS will set an internal 'RX prepared' flag when the PREPARERX task is triggered.

When the write command is received, TWIS 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, TWIS will stretch the master's clock until the PREPARERX task is triggered and the internal 'RX prepared' flag is set.

TWIS will generate the RXSTARTED event and clear the internal 'RX prepared' flag ('unprepare RX') when it enters the RX state. In this state, TWIS will be able to receive the bytes sent by the TWI master.

TWIS will go back to the IDLE state if TWIS receives a restart command when it is in the RX state.

TWIS is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. TWIS 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 RXD.PTR register. TWIS 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 it can receive, the extra bytes are discarded and 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.

TWIS can be forced to stop by triggering the STOP task. A STOPPED event will be generated when TWIS has stopped. TWIS 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 653.

TWIS 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 TWIS write command response is illustrated in the following figure, including clock stretching following a SUSPEND task.

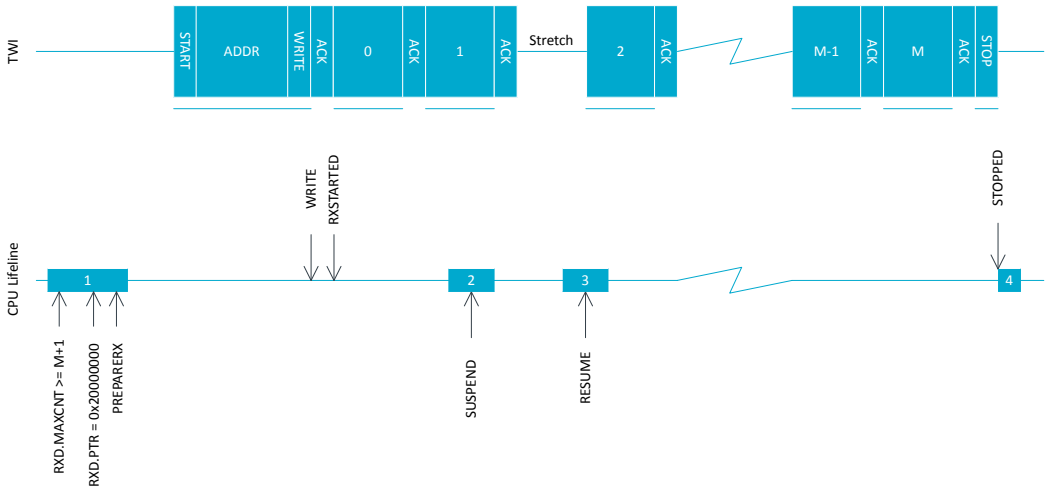

Figure 226: TWIS responding to a write command

7.37.5 Master repeated start sequence

An example of a repeated start sequence is one in which the TWI master writes two bytes to TWIS followed by reading four bytes from the slave.

This is illustrated in the following figure.

In this example, the receiver does not know what the master wants to read in advance. This information is in the first two received bytes of the write in the repeated start sequence. To guarantee that the CPU is able to process the received data before TWIS 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.

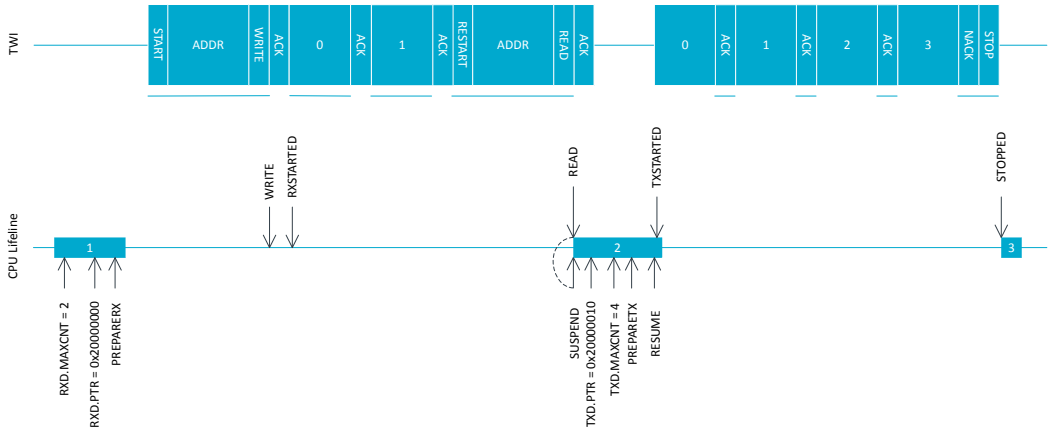

Figure 227: Repeated start sequence

7.37.6 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.

7.37.7 Low power

To ensure lowest possible power consumption when the peripheral is not needed stop and disable TWIS.

The STOP task may not be always needed (the peripheral might already be stopped), but if the task is triggered, software shall wait until the STOPPED event is generated before disabling the peripheral through the ENABLE register.

7.37.8 Slave mode pin configuration

The SCL and SDA signals are mapped to physical pins using the PSEL.SCL and PSEL.SDA registers.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as TWIS is enabled, and retained only as long as the device is in System 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 TWIS is disabled.

To secure correct signal levels on the pins used by TWIS while in System OFF mode, and when TWIS is disabled, these pins must be configured in the GPIO peripheral as described in the following table.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI slave signal	TWI slave pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 171: GPIO configuration before enabling peripheral

7.37.9 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	APPLICATION	TWIS	TWIS0 : S TWIS0 : NS	US	SA	Two-wire interface slave 0	
0x50009000 0x40009000	APPLICATION	TWIS	TWIS1 : S TWIS1 : NS	US	SA	Two-wire interface slave 1	
0x5000B000 0x4000B000	APPLICATION	TWIS	TWIS2 : S TWIS2 : NS	US	SA	Two-wire interface slave 2	
0x5000C000 0x4000C000	APPLICATION	TWIS	TWIS3 : S TWIS3 : NS	US	SA	Two-wire interface slave 3	
0x41013000	NETWORK	TWIS	TWIS0	NS	NA	Two-wire interface slave 0	

Table 172: Instances

Register	Offset	Security	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
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
SUBSCRIBE_PREPARERX	0x0B0		Subscribe configuration for task PREPARERX
SUBSCRIBE_PREPARETX	0x0B4		Subscribe configuration for task PREPARETX
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
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_WRITE	0x1E4		Publish configuration for event WRITE
PUBLISH_READ	0x1E8		Publish configuration for event READ
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4D0		Error source
MATCH	0x4D4		Status register indicating which address had a match
ENABLE	0x500		Enable TWIS
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal

Register	Offset	Security	Description
RXD.PTR	0x534		RXD Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in RXD buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last RXD transaction
RXD.LIST	0x540		EasyDMA list type
TXD.PTR	0x544		TXD Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in TXD buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last TXD transaction
TXD.LIST	0x550		EasyDMA list type
ADDRESS[n]	0x588		TWI slave address n
CONFIG	0x594		Configuration register for the address match mechanism
ORC	0x5C0		Over-read character. Character sent out in case of an over-read of the transmit buffer.

Table 173: Register overview

7.37.9.1 TASKS_STOP

Address offset: 0x014

Stop TWI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop TWI transaction
			Trigger	1	Trigger task

7.37.9.2 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

7.37.9.3 TASKS_RESUME

Address offset: 0x020

Resume TWI 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

7.37.9.4 TASKS_PREPARERX

Address offset: 0x030

Prepare the TWI slave to respond to a write command

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_PREPARERX			Prepare the TWI slave to respond to a write command
			Trigger	1	Trigger task

7.37.9.5 TASKS_PREPARETX

Address offset: 0x034

Prepare the TWI slave to respond to a read command

Bit number					31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_PREPARETX			Prepare the TWI slave to respond to a read command
			Trigger	1	Trigger task

7.37.9.6 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.37.9.7 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					B A A A A A A A A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.37.9.8 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.37.9.9 SUBSCRIBE_PREPARERX

Address offset: 0x0B0

Subscribe configuration for task PREPARERX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task PREPARERX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.37.9.10 SUBSCRIBE_PREPARETX

Address offset: 0x0B4

Subscribe configuration for task PREPARETX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task PREPARETX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.37.9.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.13 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.14 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.15 EVENTS_WRITE

Address offset: 0x164

Write command received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_WRITE			Write command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.16 EVENTS_READ

Address offset: 0x168

Read command received

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_READ			Read command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.37.9.17 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.18 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.19 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.20 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.21 PUBLISH_WRITE

Address offset: 0x1E4

Publish configuration for event WRITE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event WRITE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.22 PUBLISH_READ

Address offset: 0x1E8

Publish configuration for event READ

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event READ will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.37.9.23 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	WRITE_SUSPEND			Shortcut between event WRITE and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	READ_SUSPEND			Shortcut between event READ and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.37.9.24 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	WRITE			Enable or disable interrupt for event WRITE
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	READ			Enable or disable interrupt for event READ
			Disabled	0	Disable
			Enabled	1	Enable

7.37.9.25 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	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to enable interrupt for event WRITE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to enable interrupt for event READ
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.37.9.26 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					H G F E B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			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				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	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to disable interrupt for event WRITE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to disable interrupt for event READ
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.37.9.27 ERRORSRC

Address offset: 0x4D0

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	OVERFLOW			RX buffer overflow detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred
B	RW	DNACK			NACK sent after receiving a data byte
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	OVERREAD			TX buffer over-read detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred

7.37.9.28 MATCH

Address offset: 0x4D4

Status register indicating which address had a match

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	MATCH		[0..1]	Indication of which address in {ADDRESS} that matched the incoming address

7.37.9.29 ENABLE

Address offset: 0x500

Enable TWIS

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIS
			Disabled	0	Disable TWIS
			Enabled	9	Enable TWIS

7.37.9.30 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.37.9.31 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..1]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

7.37.9.32 RXD.PTR

Address offset: 0x534

RXD Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			RXD Data pointer
					See the memory chapter for details about which memories are available for EasyDMA.

7.37.9.33 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in RXD buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in RXD buffer

7.37.9.34 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last RXD transaction

7.37.9.35 RXD.LIST

Address offset: 0x540

EasyDMA list type

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.37.9.36 TXD.PTR

Address offset: 0x544

TXD Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			TXD Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.37.9.37 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in TXD buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in TXD buffer

7.37.9.38 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last TXD transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last TXD transaction

7.37.9.39 TXD.LIST

Address offset: 0x550

EasyDMA list type

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

ID	R/W	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

7.37.9.40 ADDRESS[n] (n=0..1)

Address offset: 0x588 + (n × 0x4)

TWI slave address n

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

7.37.9.41 CONFIG

Address offset: 0x594

Configuration register for the address match mechanism

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A-B	RW	ADDRESS[i] (i=0..1)			Enable or disable address matching on ADDRESS[i]
			Disabled	0	Disabled
			Enabled	1	Enabled

7.37.9.42 ORC

Address offset: 0x5C0

Over-read character. Character sent out in case of an over-read of the transmit buffer.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character sent out in case of an over-read of the transmit buffer.

7.37.10 Electrical specification

7.37.10.1 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIS,SCL}$	Bit rates for TWIS[29]	100		400	kbps
$t_{TWIS,START}$	Time from PREPARERX/PREPARETX task to ready to receive/ transmit		1.5		µs
t_{TWIS,SU_DAT}	Data setup time before positive edge on SCL – all modes	20			ns
t_{TWIS,HD_DAT}	Data hold time after negative edge on SCL – all modes	350		600	ns
$t_{TWIS,HD\_STA,100kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 100 kbps	500			ns
$t_{TWIS,HD\_STA,400kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 400 kbps	500			ns
$t_{TWIS,SU\_STO,100kbps}$	TWI slave setup time from SCL high to STOP condition, 100 kbps	500			ns

[29] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{TWIS,SU\_STO,400kbps}$	TWI slave setup time from SCL high to STOP condition, 400 kbps	500			ns
$t_{TWIS,BUF,100kbps}$	TWI slave bus free time between STOP and START conditions, 100 kbps	500			ns
$t_{TWIS,BUF,400kbps}$	TWI slave bus free time between STOP and START conditions, 400 kbps	500			ns

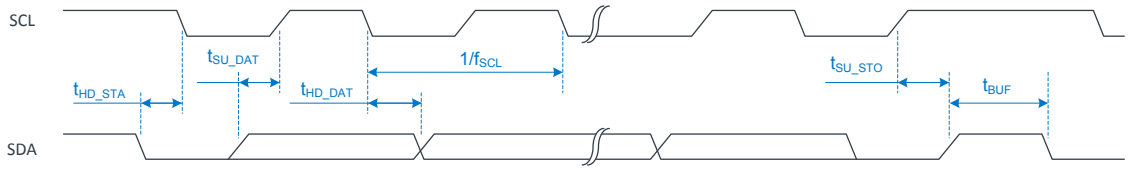

Figure 228: TWIS timing diagram, 1 byte transaction

7.38 UARTE — Universal asynchronous receiver/transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA (UARTE) offers fast, full-duplex, asynchronous serial communication. Built-in flow control (CTS, RTS) is supported in hardware at a rate up to 1 Mbps and EasyDMA data transfer to and from RAM.

The main features of UARTE are the following:

- Full-duplex operation
- Automatic hardware flow control
- Optional even parity bit checking and generation
- EasyDMA
- Up to 1 Mbps baudrate
- Return to IDLE between transactions supported (when using HW flow control)
- One or two stop bit
- Least significant bit (LSB) first

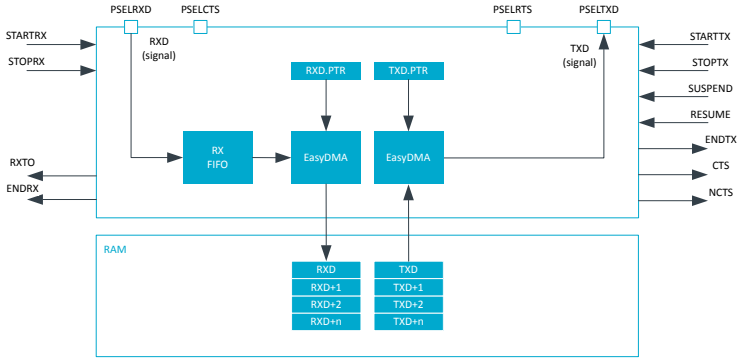

Figure 229: UARTE configuration

The GPIOs used for each UART interface can be chosen from any GPIO on the device and are independently configurable. This enables device pinout flexibility and efficient use of board space and signal routing.

Note: The external crystal oscillator must be enabled to obtain sufficient clock accuracy for stable communication. See CLOCK — Clock control on page 69 for more information.

7.38.1 EasyDMA

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 18 for more information about the different memory regions.

The RXD.PTR, TXD.PTR, RXD.MAXCNT, and TXD.MAXCNT registers are double-buffered. They can be updated and prepared for the next reception or transmission immediately after having received the RXSTARTED or TXSTARTED event.

The ENDRX and ENDTX events indicate that the EasyDMA is finished accessing the RX or TX buffer in RAM.

7.38.2 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to TXD.PTR, and the number of bytes to transmit from 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 have been transmitted, the transmission will automatically end and the 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, UARTE will generate the ENDTX event explicitly even though all bytes specified in the TXD.MAXCNT register have not been transmitted.

If flow control is enabled through the HWFC field in the CONFIG register, a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as shown in the following figure. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended.

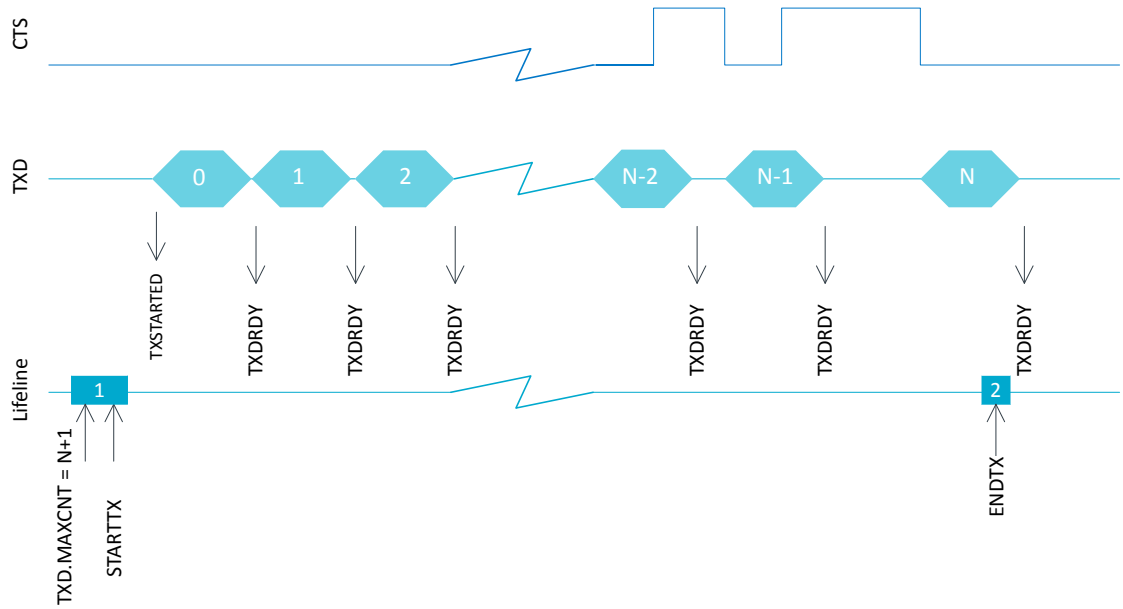

Figure 230: UARTE transmission

The UARTE transmitter is in its lowest activity level consuming the least amount of energy when it is stopped. That is, before it is started via STARTTX or after it has been stopped via STOPTX and the

TXSTOPPED event has been generated. See POWER — Power control on page 43 for more information about power modes.

7.38.3 Reception

The UARTE receiver is started by triggering the STARTRX task. The UARTE receiver uses 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. UARTE generates an ENDRX event when it has filled up the RX buffer, as seen in the following figure.

For each byte received over the RXD line, an RXDRDY event is 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.

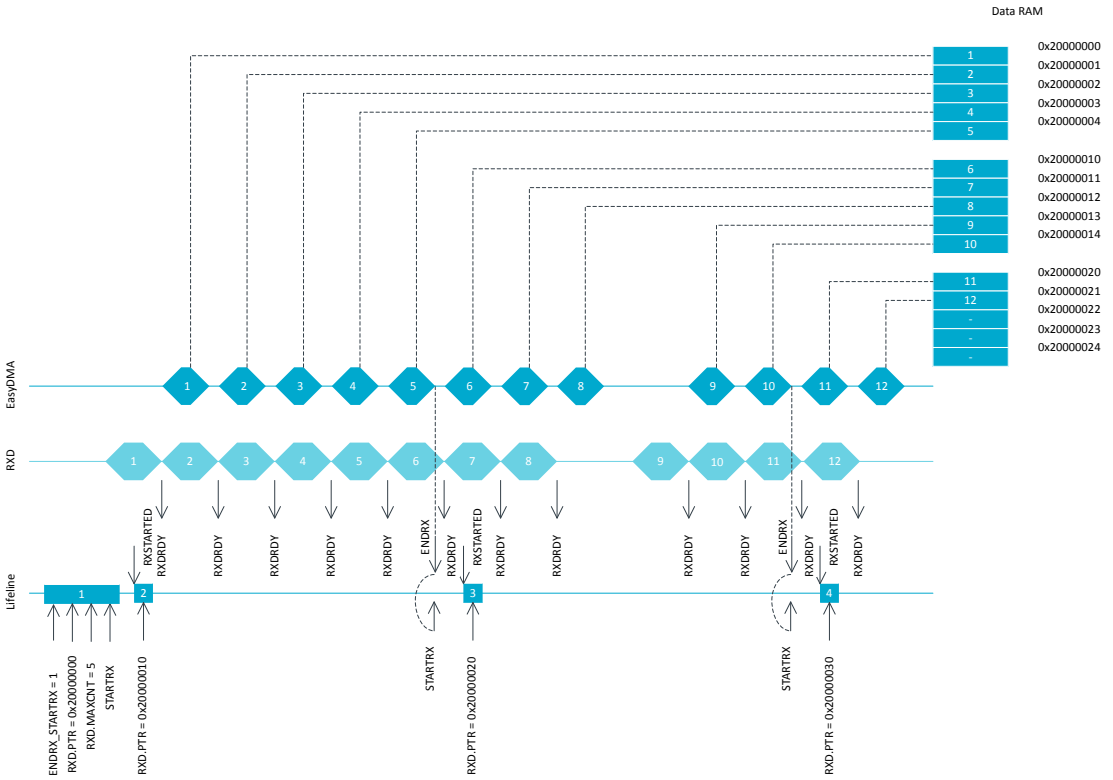

Figure 231: UARTE reception

The UARTE receiver is stopped by triggering the STOPRX task. An RXTO event is generated when the UARTE has stopped. UARTE makes sure that an impending ENDRX event is generated before the RXTO event is generated. This means that UARTE guarantees that no ENDRX event is generated after RXTO, unless UARTE is restarted or a FLUSHRX command is issued after the RXTO event is generated.

Note: If the ENDRX event has not been generated when the UARTE receiver stops, indicating that all pending content in the RX FIFO has been moved to the RX buffer, UARTE generates the ENDRX event explicitly even though the RX buffer is not full. In this scenario the ENDRX event is generated before the RXTO event is generated.

To determine the amount of bytes the RX buffer has received, the CPU can read the RXD.AMOUNT register following the ENDRX event or the RXTO event.

UARTE can receive up to four bytes after the STOPRX task has been triggered, if these are sent in succession immediately after the RTS signal is deactivated.

After the RXTO event is generated, the internal RX FIFO may still contain data. To move this data to RAM, the FLUSHRX task must be triggered. The RX buffer should be emptied, or the RXD.PTR register should be updated before the FLUSHRX task is triggered. This ensures the data in the RX buffer is not overwritten. 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, as seen in the following figure. 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 fill up. After the ENDRX event, the RXD.AMOUNT register holds the actual amount of bytes transferred to the RX buffer.

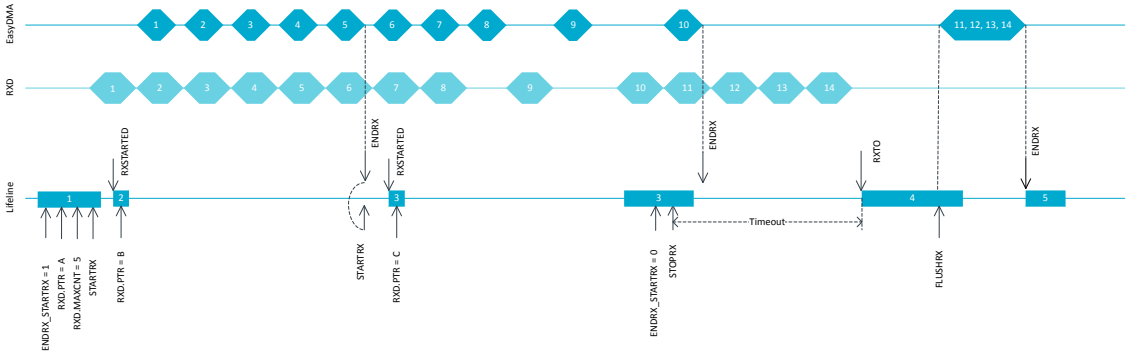

Figure 232: UARTE reception with forced stop via STOPRX

If hardware flow control is enabled through the HWFC field in the CONFIG register, the RTS signal will be deactivated when the receiver is stopped via the STOPRX task or when the UARTE is only able to receive four more bytes in its internal RX FIFO.

With flow control disabled, the UARTE will function in the same way as when the flow control is enabled except that the RTS line will not be used. This means that no signal will be generated when the UARTE has reached the point where it is only able to receive four more bytes in its internal RX FIFO. Data received when the internal RX FIFO is filled up, will be lost.

The UARTE receiver will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTRX or after it has been stopped via STOPRX and the RXTO event has been generated. See POWER — Power control on page 43 for more information about power modes.

7.38.4 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

An ERROR event will not stop reception. If the error was a parity error, the received byte is still transferred into Data RAM along with any following bytes. If a framing error occurs (wrong stop bit), that byte will not be stored into Data RAM but following incoming bytes will.

7.38.5 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

7.38.6 Parity and stop bit configuration

Automatic even parity generation for both transmission and reception can be configured using the register CONFIG on page 690. If odd parity is desired, it can be configured using the register CONFIG on page 690. See the register description for details.

The amount of stop bits can also be configured through the register CONFIG on page 690.

7.38.7 Low power

To ensure lowest possible power consumption when the peripheral is not needed, stop and disable UARTE.

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.

7.38.8 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with 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 System 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 in System OFF mode, the pins must be configured in the GPIO peripheral as described in the following table.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

UARTE signal	UARTE pin	Direction	Output value
RXD	As specified in PSEL.RXD	Input	Not applicable
CTS	As specified in PSEL.CTS	Input	Not applicable
RTS	As specified in PSEL.RTS	Output	1
TXD	As specified in PSEL.TXD	Output	1

Table 174: GPIO configuration before enabling peripheral

7.38.9 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	APPLICATION	UARTE	UARTE0 : S UARTE0 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 0	
0x50009000 0x40009000	APPLICATION	UARTE	UARTE1 : S UARTE1 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 1	
0x5000B000 0x4000B000	APPLICATION	UARTE	UARTE2 : S UARTE2 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 2	
0x5000C000 0x4000C000	APPLICATION	UARTE	UARTE3 : S UARTE3 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 3	
0x41013000	NETWORK	UARTE	UARTE0	NS	NA	Universal asynchronous receiver/transmitter	

Table 175: Instances

Register	Offset	Security	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
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STOPRX	0x084		Subscribe configuration for task STOPRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOPTX	0x08C		Subscribe configuration for task STOPTX
SUBSCRIBE_FLUSHRX	0x0AC		Subscribe configuration for task FLUSHRX
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
PUBLISH_CTS	0x180		Publish configuration for event CTS
PUBLISH_NCTS	0x184		Publish configuration for event NCTS
PUBLISH_RXDRDY	0x188		Publish configuration for event RXDRDY
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_TXDRDY	0x19C		Publish configuration for event TXDRDY
PUBLISH_ENDTX	0x1A0		Publish configuration for event ENDTX
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXTO	0x1C4		Publish configuration for event RXTO
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_TXSTOPPED	0x1D8		Publish configuration for event TXSTOPPED
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x480		Error source
ENABLE	0x500		Enable UART
PSEL.RTS	0x508		Pin select for RTS signal
PSEL.TXD	0x50C		Pin select for TXD signal
PSEL.CTS	0x510		Pin select for CTS signal
PSEL.RXD	0x514		Pin select for RXD signal
BAUDRATE	0x524		Baud rate. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
CONFIG	0x56C		Configuration of parity and hardware flow control

Table 176: Register overview

7.38.9.1 TASKS_STARTRX

Address offset: 0x000

Start UART receiver

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description	
A	W	TASKS_STARTRX			Start UART receiver	
			Trigger	1	Trigger task	

7.38.9.2 TASKS_STOPRX

Address offset: 0x004

Stop UART receiver

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description	
A	W	TASKS_STOPRX			Stop UART receiver	
			Trigger	1	Trigger task	

7.38.9.3 TASKS_STARTTX

Address offset: 0x008

Start UART transmitter

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description	
A	W	TASKS_STARTTX			Start UART transmitter	
			Trigger	1	Trigger task	

7.38.9.4 TASKS_STOPTX

Address offset: 0x00C

Stop UART transmitter

Bit number						31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description	
A	W	TASKS_STOPTX			Stop UART transmitter	
			Trigger	1	Trigger task	

7.38.9.5 TASKS_FLUSHRX

Address offset: 0x02C

Flush RX FIFO into RX 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_FLUSHRX			Flush RX FIFO into RX buffer
			Trigger	1	Trigger task

7.38.9.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.38.9.7 SUBSCRIBE_STOPRX

Address offset: 0x084

Subscribe configuration for task STOPRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOPRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.38.9.8 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.38.9.9 SUBSCRIBE_STOPTX

Address offset: 0x08C

Subscribe configuration for task STOPTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOPTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.38.9.10 SUBSCRIBE_FLUSHRX

Address offset: 0x0AC

Subscribe configuration for task FLUSHRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task FLUSHRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.38.9.11 EVENTS_CTS

Address offset: 0x100

CTS is activated (set low). Clear To Send.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_CTS			CTS is activated (set low). Clear To Send.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.12 EVENTS_NCTS

Address offset: 0x104

CTS is deactivated (set high). Not Clear To Send.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_NCTS			CTS is deactivated (set high). Not Clear To Send.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.13 EVENTS_RXDRDY

Address offset: 0x108

Data received in RXD (but potentially not yet transferred to Data RAM)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXDRDY			Data received in RXD (but potentially not yet transferred to Data RAM)
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.14 EVENTS_ENDRX

Address offset: 0x110

Receive buffer is filled up

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			Receive buffer is filled up
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.15 EVENTS_TXDRDY

Address offset: 0x11C

Data sent from TXD

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXDRDY			Data sent from TXD
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.16 EVENTS_ENDTX

Address offset: 0x120

Last TX byte transmitted

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			Last TX byte transmitted
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.17 EVENTS_ERROR

Address offset: 0x124

Error detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			Error detected
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.18 EVENTS_RXTO

Address offset: 0x144

Receiver timeout

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXTO			Receiver timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.19 EVENTS_RXSTARTED

Address offset: 0x14C

UART receiver has started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			UART receiver has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.20 EVENTS_TXSTARTED

Address offset: 0x150

UART transmitter has started

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			UART transmitter has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.21 EVENTS_TXSTOPPED

Address offset: 0x158

Transmitter stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTOPPED			Transmitter stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.38.9.22 PUBLISH_CTS

Address offset: 0x180

Publish configuration for event CTS

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event CTS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.23 PUBLISH_NCTS

Address offset: 0x184

Publish configuration for event NCTS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event NCTS will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.24 PUBLISH_RXDRDY

Address offset: 0x188

Publish configuration for event RXDRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXDRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.25 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDRX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.26 PUBLISH_TXDRDY

Address offset: 0x19C

Publish configuration for event TXDRDY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXDRDY will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.27 PUBLISH_ENDTX

Address offset: 0x1A0

Publish configuration for event ENDTX

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.28 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.29 PUBLISH_RXTO

Address offset: 0x1C4

Publish configuration for event RXTO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXTO will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.30 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.31 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.32 PUBLISH_TXSTOPPED

Address offset: 0x1D8

Publish configuration for event TXSTOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TXSTOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.38.9.33 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

ID	R/W	Field	Value ID	Value	Description
C	RW	ENDRX_STARTRX			Shortcut between event ENDRX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDRX_STOPRX			Shortcut between event ENDRX and task STOPRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.38.9.34 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					L J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CTS			Enable or disable interrupt for event CTS
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	NCTS			Enable or disable interrupt for event NCTS
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	RXDRDY			Enable or disable interrupt for event RXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ENDRX			Enable or disable interrupt for event ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXDRDY			Enable or disable interrupt for event TXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	ENDTX			Enable or disable interrupt for event ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	RXTO			Enable or disable interrupt for event RXTO
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	TXSTOPPED			Enable or disable interrupt for event TXSTOPPED
			Disabled	0	Disable
			Enabled	1	Enable

7.38.9.35 INTENSET

Address offset: 0x304

Enable interrupt

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

ID	R/W	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to enable interrupt for event RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to enable interrupt for event ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to enable interrupt for event TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to enable interrupt for event ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to enable interrupt for event RXTO
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to enable interrupt for event TXSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.38.9.36 INTENCLR

Address offset: 0x308

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to disable interrupt for event CTS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to disable interrupt for event NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to disable interrupt for event RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to disable interrupt for event TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to disable interrupt for event ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to disable interrupt for event RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to disable interrupt for event TXSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.38.9.37 ERRORSRC

Address offset: 0x480

Error source

This register is read/write one to clear.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A start bit is received while the previous data still lies in RXD. (Previous data is lost.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
B	RW	PARITY			Parity error A character with bad parity is received, if HW parity check is enabled.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
C	RW	FRAMING			Framing error occurred A valid stop bit is not detected on the serial data input after all bits in a character have been received.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
D	RW	BREAK			Break condition The serial data input is '0' for longer than the length of a data frame. (The data frame length is 10 bits without parity bit and 11 bits with parity bit.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present

7.38.9.38 ENABLE

Address offset: 0x500

Enable UART

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

ID	R/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable UARTE
			Disabled	0	Disable UARTE
			Enabled	8	Enable UARTE

7.38.9.39 PSEL.RTS

Address offset: 0x508

Pin select for RTS signal

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

7.38.9.40 PSEL.TXD

Address offset: 0x50C

Pin select for TXD signal

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

7.38.9.41 PSEL.CTS

Address offset: 0x510

Pin select for CTS signal

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

7.38.9.42 PSEL.RXD

Address offset: 0x514

Pin select for RXD signal

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

7.38.9.43 BAUDRATE

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	BAUDRATE			Baud rate
			Baud1200	0x0004F000	1200 baud (actual rate: 1205)
			Baud2400	0x0009D000	2400 baud (actual rate: 2396)
			Baud4800	0x0013B000	4800 baud (actual rate: 4808)
			Baud9600	0x00275000	9600 baud (actual rate: 9598)
			Baud14400	0x003AF000	14400 baud (actual rate: 14401)
			Baud19200	0x004EA000	19200 baud (actual rate: 19208)
			Baud28800	0x0075C000	28800 baud (actual rate: 28777)
			Baud31250	0x00800000	31250 baud
			Baud38400	0x009D0000	38400 baud (actual rate: 38369)
			Baud56000	0x00E50000	56000 baud (actual rate: 55944)
			Baud57600	0x00EB0000	57600 baud (actual rate: 57554)
			Baud76800	0x013A9000	76800 baud (actual rate: 76923)
			Baud115200	0x01D60000	115200 baud (actual rate: 115108)
			Baud230400	0x03B00000	230400 baud (actual rate: 231884)
			Baud250000	0x04000000	250000 baud
			Baud460800	0x07400000	460800 baud (actual rate: 457143)
			Baud921600	0x0F000000	921600 baud (actual rate: 941176)
			Baud1M	0x10000000	1 megabaud

7.38.9.44 RXD.PTR

Address offset: 0x534

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.38.9.45 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in receive buffer

7.38.9.46 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

7.38.9.47 TXD.PTR

Address offset: 0x544

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the Memory chapter for details about which memories are available for EasyDMA.

7.38.9.48 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in transmit buffer

7.38.9.49 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0xFFFF]	Number of bytes transferred in the last transaction

7.38.9.50 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				D C B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	HWFC			Hardware flow control
			Disabled	0	Disabled
			Enabled	1	Enabled
B	RW	PARITY			Parity
			Excluded	0x0	Exclude parity bit
			Included	0x7	Include even parity bit
C	RW	STOP			Stop bits
			One	0	One stop bit
			Two	1	Two stop bits
D	RW	PARITYTYPE			Even or odd parity type
			Even	0	Even parity
			Odd	1	Odd parity

7.38.10 Electrical specification

7.38.10.1 UARTE electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
f_{UARTE}	Baud rate for UARTE[30].			1000	kbps
$t_{UARTE,CTSH}$	CTS high time	0.5			µs
$t_{UARTE,START}$	Time from STARTRX/STARTTX task to transmission started		0.25		µs

7.39 USBD — Universal serial bus device

The USB device (USBD) controller implements a full speed USB device function that meets 2.0 revision of the USB specification.

[30] High baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

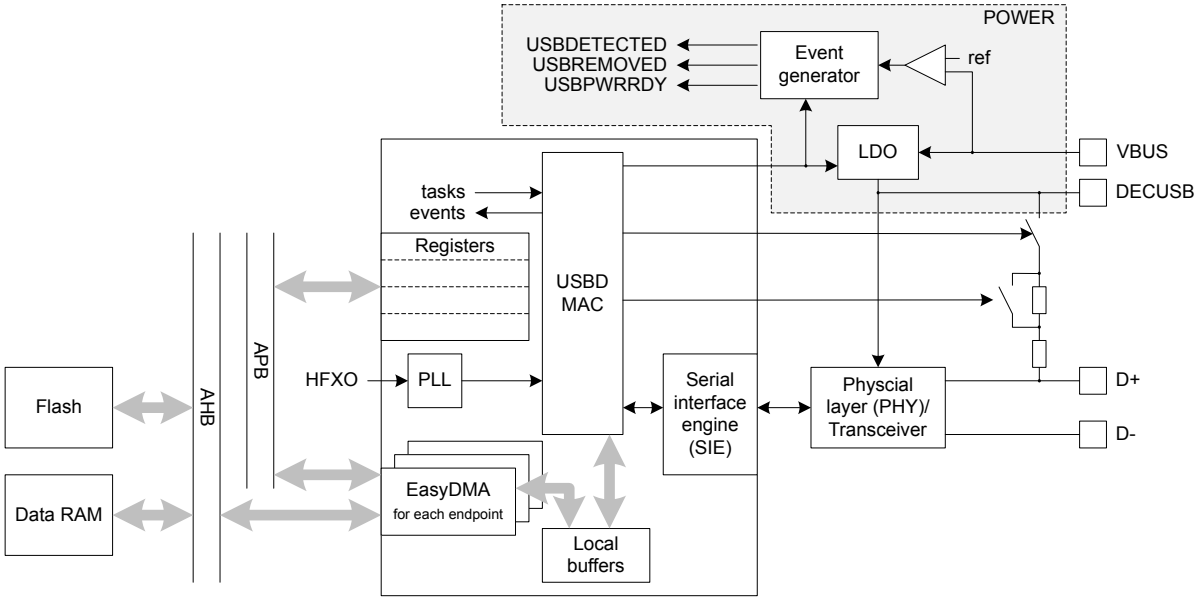

Figure 233: USB device block diagram

Listed here are the main features for USBD:

- Implements full-speed (12 Mbps) device fully compliant to Universal Serial Bus Specification Revision 2.0, including following engineering change notices (ECNs) issued by USB Implementers Forum:
  - *Pull-up/pull-down Resistors ECN*
  - *5V Short Circuit Withstand Requirement Change ECN*
- USB device stack available in the Nordic SDK
- Integrated (on-chip) USB transceiver (PHY)
- Software controlled on-chip pull-up on D+
- Endpoints:
  - 2 control (1 IN, 1 OUT)
  - 14 bulk/interrupt (7 IN, 7 OUT)
  - 2 isochronous (1 IN, 1 OUT)
- Supports double buffering for isochronous (ISO) endpoints (IN/OUT)
- Supports USB suspend, resume, and remote wake-up
- 64 bytes buffer size for each bulk/interrupt endpoint
- Up to 1023 bytes buffer size for ISO endpoints
- EasyDMA for all data transfers

7.39.1 USB device states

The behavior of a USB device can be modelled through a state diagram.

The *USB 2.0 Specification* (see *Chapter 9 USB Device Framework*) defines a number of states for a USB device, as shown in the following figure.

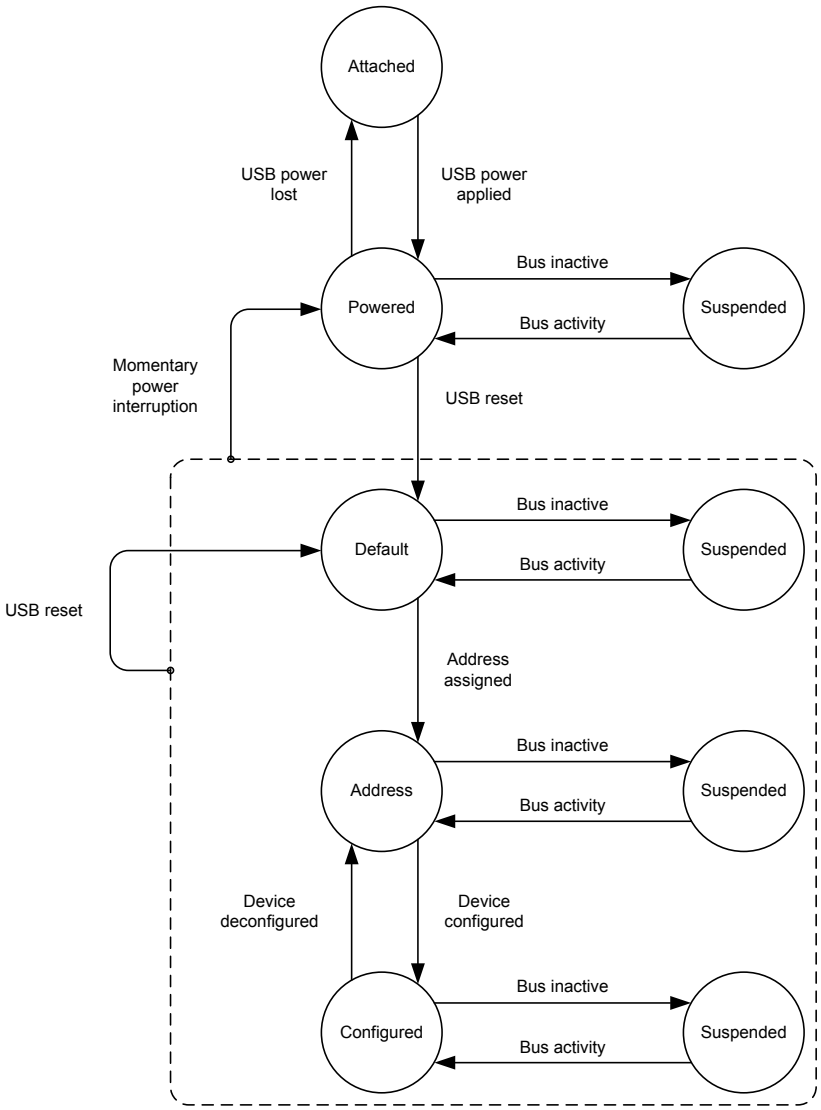

Figure 234: Device state diagram

The device must change state according to host-initiated traffic and USB bus states. It is up to the software to implement a state machine that matches the above definition. To detect the presence or absence of USB supply (VBUS), two events USBDETECTED and USBREMOVED can be used to implement the state machine. For more details on these events, see USBREG — USB regulator control on page 55.

As a general rule when implementing the software, the host behavior shall never be assumed to be predictable. In particular the sequence of commands received during an enumeration. The software shall always react to the current bus conditions or commands sent by the host.

7.39.2 USB terminology

The USB specification defines bus states, rather than logic levels on the D+ and D- lines.

For a full speed device, the bus state where the D+ line is high and the D- line is low is defined as the J state. The bus state where D+ is low and D- high is called the K state.

An idle bus, where D+ and D- lines are only polarized through the pull-up on D+ and pull-downs on the host side, will be in J state.

Both lines low are called SE0 (single-ended 0), and both lines high SE1 (single-ended 1).

7.39.3 USB pins

The USBD peripheral features a number of dedicated pins.

The dedicated USB pins can be grouped in two categories, signal and power. The signal pins consist of the D+ and D- pins, which are to be connected to the USB host. They are dedicated pins, and not available as standard GPIOs. The USBD peripheral is implemented according to the USB specification revision 2.0, *5V Short Circuit Withstand ECN Requirement Change*, meaning these two pins are not 5 V tolerant.

The signal pins and the pull-up will operate only while VBUS is in its valid voltage range, and USBD is enabled through the ENABLE register. For details on the USB power supply and VBUS detection, see USBREG — USB regulator control on page 55.

For more information about the pinout, see Pin assignments on page 783.

7.39.4 USBD power-up sequence

The physical layer interface (PHY)/USB transceiver is powered separately from the rest of the device (VBUS pin), which has some implications on the USBD power-up sequence.

The device is not able to properly signal its presence to the USB host and handle traffic from the host, unless the PHY's power supply is enabled and stable. Turning the PHY's power supply on/off is directly linked to register ENABLE. The device provides events that help synchronizing software to the various steps during the power-up sequence.

To make sure that all resources in USBD are available and the dedicated USB voltage regulator stabilized, the following is recommended:

- Enable USBD only after VBUS has been detected
- Turn the USB pull-up on after the following events have occurred:
  - USBPWRRDY
  - USBEVENT, with the READY condition flagged in EVENTCAUSE

The following sequence chart illustrates a typical handling of VBUS power-up:

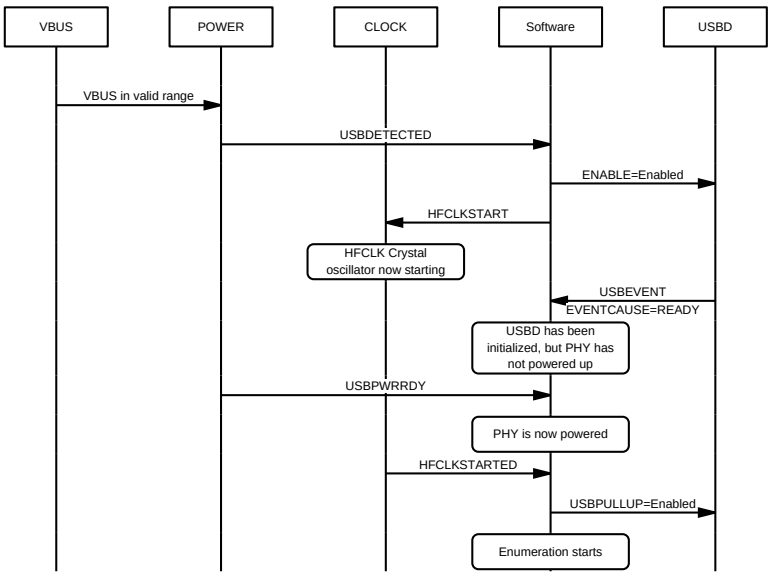

Figure 235: VBUS power-up sequence

Upon detecting VBUS removal, it is recommended to wait for ongoing EasyDMA transfers to finish before disabling USBD (relevant ENDEPIN[n], ENDISOIN, ENDEPOUT[n], or ENDISOOUT events, see EasyDMA on page 696). The USBREMOVED event, described in USBREG — USB regulator control on page 55, signals when the VBUS is removed. Reading the ENABLE register will return Enabled until USBD is completely disabled.

7.39.5 USB pull-up

The USB pull-up serves two purposes: it indicates to the host that the device is connected to the USB bus, and it indicates the device's speed capability.

When no pull-up is connected to the USB bus, the host sees both D+ and D- lines low, as they are pulled down on the host side by 15 kΩ resistors. The device is not detected by the host, putting it in a detached state even if it is physically connected to the host. In this situation, the device is not allowed to draw current from VBUS, according to *USB 2.0 Specification*.

When a full-speed device connects its 1.5 kΩ pull-up to D+, the host sees the corresponding line high. The device is then in the attached state. During the enumeration process, the host attempts to determine if the full-speed device also supports higher speeds and initiates communication with the device to further identify it. The USBD peripheral implemented in this device supports only full-speed operation (12 Mbps), and thus ignores the negotiation for higher speeds in accordance with *USB 2.0 Specification*.

Register USBPULLUP enables software to connect or disconnect the pull-up on D+. This allows the software to control when USB enumeration takes place. It also allows to emulate a physical disconnect from the USB bus, for instance when re-enumeration is required. USBPULLUP has to be enabled to allow the USBD to handle USB traffic and generate appropriate events. This forbids the use of an external pull-up.

Note that disconnecting the pull-up through register USBPULLUP while connected to a host, will result in both D+ and D- lines to be pulled low by the host's pull-down resistors. However, as mentioned above, this will also inhibit the generation of the USBRESET event. The pull-up is disabled by default after a chip reset.

The pull-up shall only get connected after USBD has been enabled through register ENABLE. The USB pull-up value is automatically changed depending on the bus activity, as specified in *Resistor ECN* which amends the original *USB 2.0 Specification*. The user does not have access to this function as it is handled in hardware.

While they should never be used in normal traffic activity, lines D+ and D- may at any time be forced into state specified in register DPDMVALUE by the task DPDMDRIVE. The DPDMNODRIVE task stops driving them, and PHY returns to normal operation.

7.39.6 USB reset

The USB specification defines a USB reset, which is not be confused with a chip reset. The USB reset is a normal USB bus condition, and is used as part of the enumeration sequence, it does not reset the chip.

The USB reset results from a single-ended low state (SE0) on lines D+/D- for a $t_{USB,DETRST}$ amount of time. Only the host is allowed to drive a USB reset condition on the bus. The UBSD peripheral automatically interprets a SE0 longer than $t_{USB,DETRST}$ as a USB reset. When the device detects a USB reset and generates a USBRESET event, the device USB stack and related parts of the application shall re-initialize themselves, and go back to the default state.

Some of the registers in the USBD peripheral get automatically reset to a known state, in particular all data endpoints are disabled and the USBADDR reset to 0.

After the device has connected to the USB bus (i.e. after VBUS is applied), the device shall not respond to any traffic from the time the pull-up is enabled until it has seen a USB reset condition. This is automatically ensured by the USBD.

After a USB reset, the device shall be fully responsive after at most T_{RSTRCY} (according to chapter 7 in the USB specification). Software shall take into account this time that takes the hardware to recover from a USB reset condition.

7.39.7 USB suspend and resume

Normally, the host will maintain activity on the USB at least every millisecond according to USB specification. A USB device will enter suspend when there is no activity on the bus (idle) for a given time. The device will resume operation when it receives any non idle signalling.

To signal that the device shall go into low power mode (suspend), the host stops activity on the USB bus, which becomes idle. Only the device pull-up and host pull-downs act on D+ and D-, and the bus is thus kept at a constant J state. It is up to the device to detect this lack of activity, and enter the low power mode (suspend) within a specified time.

The USB host can decide to suspend or resume USB activity at any time. If remote wake-up is enabled, the device may signal to the host to resume from suspend.

7.39.7.1 Entering suspend

The USBD peripheral automatically detects lack of activity for more than a defined amount of time, and performs steps needed to enter suspend.

When no activity has been detected for longer than $t_{USB,SUSPEND}$, the USBD generates the USBEVENT event with SUSPEND bit set in register EVENTCAUSE. The software shall ensure that the current drawn from the USB supply line VBUS is within the specified limits before T_{2SUSP}, as defined in chapter 7 of the USB specification. In order to reduce idle current of USBD, the software must explicitly place the USBD in low power mode through writing `LowPower` to register LOWPOWER.

In order to save power, and provided that no other peripheral needs it, the crystal oscillator (HFXO) in CLOCK may be disabled by software during the USB suspend, while the USB pull-up is disconnected, or when VBUS is not present. Software must explicitly enable it at any other time. The USBD will not be able to respond to USB traffic unless HFXO is enabled and stable.

7.39.7.2 Host-initiated resume

Once the host resumes the bus activity, it has to be responsive to incoming requests on the USB bus within the time T_{RSMRCY} (as defined in chapter 7 of the USB specification) and revert to normal power consumption mode.

If the host resumes bus activity with or without a RESUME condition (in other words: bus activity is defined as any non-J state), the USBD peripheral will generate a USBEVENT event, with RESUME bit set in register EVENTCAUSE. If the host resumes bus activity simply by restarting sending frames, the USBD peripheral will generate SOF events.

7.39.7.3 Device-initiated remote wake-up

Assuming the remote wake-up is supported by the device and enabled by the host, the device can request the host to resume from suspend if wake-up condition is met.

To do so, the HFXO needs to be enabled first. After waking up the HFXO, the software must bring USBD out of the low power mode and into the normal power consumption mode through writing `ForceNormal` in register LOWPOWER. It can then instruct the USBD peripheral to drive a RESUME condition (K state) on the USB bus by triggering the DPDMDRIVE task, and hence attempt to wake up the host. By choosing `Resume` in DPDMVALUE, the duration of the RESUME state is under hardware control ($t_{USB,DRIVEK}$). By choosing `J` or `K`, the duration of that state is under software control (the J or K state is maintained until a DPDMNODRIVE task is triggered) and has to meet T_{DRSMUP} as specified in USB specification chapter 7.

Upon writing the `ForceNormal` in register LOWPOWER, a USBEVENT event is generated with the USBWUALLOWED bit set in register EVENTCAUSE.

The value in register DPDMVALUE on page 728 will only be captured and used when the DPDMDRIVE task is triggered. This value defines the state the bus will be forced into after the DPDMDRIVE task.

The device shall ensure that it does not initiate a remote wake-up request before T_{WTRSM} (according to USB specification chapter 7) after the bus has entered idle state. Using the recommended resume value in DPDMVALUE (rather than K) takes care of this, and postpones the RESUME state accordingly.

7.39.8 EasyDMA

The USBD peripheral implements EasyDMA for accessing memory without CPU involvement.

Each endpoint has an associated set of registers, tasks and events. EasyDMA and traffic on USB are tightly related. A number of events provide insight of what is happening on the USB bus with a number of tasks allowing an automated response to the traffic.

> **Note:** Endpoint 0 (IN and OUT) are implemented as control endpoint. For more information, see Control transfers on page 697.

Registers

Enabling endpoints is controlled through the EPINEN and EPOUTEN registers.

The following registers define the memory address of the buffer for a specific IN or OUT endpoint:

- EPIN[n].PTR, (n=0..7)
- EPOUT[n].PTR, (n=0..7)
- ISOIN.PTR
- ISOOUT.PTR

The following registers define the amount of bytes to be sent on USB for next transaction:

- EPIN[n].MAXCNT, (n=0..7)
- ISOIN.MAXCNT

The following registers define the length of the buffer (in bytes) for next transfer of incoming data:

- EPOUT[n].MAXCNT, (n=1..7)
- ISOOUT.MAXCNT

Since the host decides how many bytes are sent over USB, the MAXCNT value can be copied from register SIZE.EPOUT[n] (n=1..7) or register SIZE.ISOOUT.

Register EPOUT[0].MAXCNT defines the length of the OUT buffer (in bytes) for the control endpoint 0. Register SIZE.EPOUT[0] shall indicate the same value as `MaxPacketSize` from the device descriptor or `wLength` from the SETUP command, whichever is the least.

The .AMOUNT registers indicate how many bytes actually have been transferred over EasyDMA during the last transfer.

Stalling bulk/interrupt endpoints is controlled through the EPSTALL register.

> **Note:** Due to USB specification requirements, the effect of the stalling control endpoint 0 may be overridden by hardware, in particular when a new SETUP token is received.

EasyDMA will not copy the SETUP data to memory (it will only transfer data from the data stage). The following are separate registers in the USBD peripheral that have setup data.

- BMREQUESTTYPE
- BREQUEST
- WVALUEL
- WVALUEH
- WINDEXL
- WINDEXH

- WLENGTHL
- WLENGTHH

The EVENTCAUSE register provides details on what caused a given USBEVENT event, for instance if a CRC error is detected during a transaction, or if bus activity stops or resumes.

Tasks

Tasks STARTEPIN[n], STARTEPOUT[n] (n=0..7), STARTISOIN, and STARTISOOUT capture the values for .PTR and .MAXCNT registers. For IN endpoints, a transaction over USB gets automatically triggered when the EasyDMA transfer is complete. For OUT endpoints, it is up to software to allow the next transaction over USB. See the examples in Control transfers on page 697, Bulk and interrupt transactions on page 700, and Isochronous transactions on page 702.

For the control endpoint 0, OUT transactions are allowed through the EP0RCVOUT task. The EP0STATUS task allows a status stage to be initiated, and the EP0STALL task allows stalling further traffic (data or status stage) on the control endpoint.

Events

The STARTED event confirms that the values of the .PTR and .MAXCNT registers of the endpoints flagged in register EPSTATUS have been captured. Those can then be modified by software for the next transfer.

Events ENDEPIN[n], ENDEPOUT[n] (n=0..7), ENDISOIN, and ENDISOOUT events indicate that the entire buffer has been consumed. The buffer can be accessed safely by the software.

Only a single EasyDMA transfer can take place in USBD at any time. Software must ensure that tasks STARTEPIN[n] (n=0..7), STARTISOIN , STARTEPOUT[n] (n=0..7), or STARTISOOUT are not triggered before events ENDEPIN[n] (n=0..7), ENDISOIN, ENDEPOUT[n] (n=0..7), or ENDISOOUT are received from an on-going transfer.

The EPDATA event indicates that a successful (acknowledged) data transaction has occurred on the data endpoint(s) flagged in register EPDATASTATUS. A successful (acknowledged) data transaction on endpoint 0 is signalled by the EP0DATADONE event.

At any time a USBEVENT event may be sent, with details provided in EVENTCAUSE register.

The EP0SETUP event indicates that a SETUP token has been received on the control endpoint 0, and that the setup data is available in #unique_1726/unique_1726_Connect_42_setup_data_registers on page 696.

7.39.9 Control transfers

The USB specification mandates every USB device to implement endpoint 0 IN and OUT as control endpoints.

A control transfer consists of two or three stages:

- Setup stage
- Data stage (optional)
- Status stage

Each control transfer can be one of following types:

- Control read
- Control read no data
- Control write
- Control write no data

An EP0SETUP event indicates that the data in the setup stage (following the SETUP token) is available in registers.

The data in the data stage (following the IN or OUT token) is transferred from or to the desired location using EasyDMA.

The control endpoint buffer can be of any size.

After receiving the SETUP token, the USB controller will not accept (NAK) any incoming IN or OUT tokens until the software has finished decoding the command, determined the type of transfer, and prepared for the next stage (data or status) appropriately.

The software can stall a command when in the data and status stages, through the EP0STALL task, when the command is not supported or if its wValue, wIndex or wLength parameters are wrong. The following shows a stalled control read transfer, but the same mechanism (tasks) applies to stalling a control write transfer.

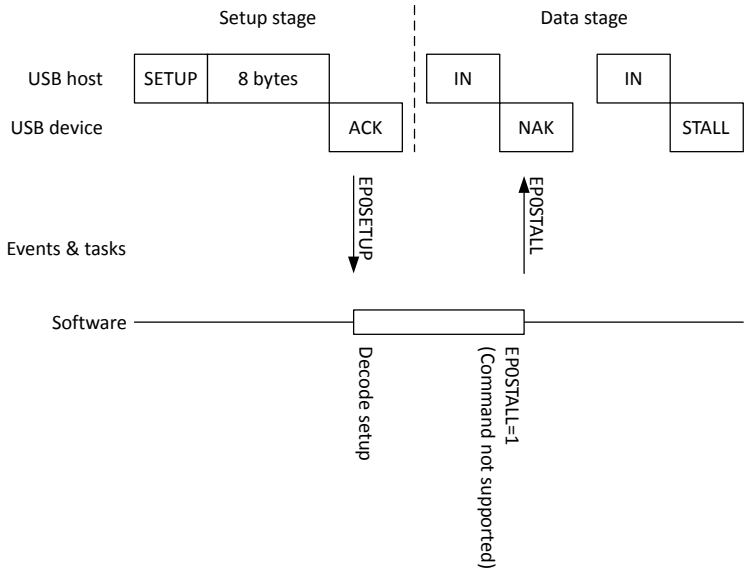

Figure 236: Control read gets stalled

See the *USB 2.0 Specification* and relevant class specifications for rules on stalling commands.

Note: The USBD peripheral handles the SetAddress transfer by itself. As a consequence, the software shall not process this command other than updating its state machine (see Device state diagram), nor initiate a status stage. If necessary, the address assigned by the host can be read out from the USBADDR register after the command has been processed.

7.39.9.1 Control read transfer

This section describes how the software behaves when responding to a control read transfer.

As mentioned earlier, the USB controller will not accept (NAK) any incoming IN tokens until software has finished decoding the command, determining the type of transfer, and preparing for the next stage (data or status) appropriately.

For a control read, transferring the data from memory into USBD will trigger a valid, acknowledged (ACK) IN transaction on USB.

The software has to prepare EasyDMA by pointing to the buffer containing the data to be transferred. If no other EasyDMA transfers are on-going with USBD, the software can send the STARTEPIN0 task, which will initiate the data transfer and transaction on USB.

A STARTED event (with EPIN0 bit set in the EPSTATUS register) will be generated as soon as the EPIN[0].PTR and .MAXCNT registers have been captured. Software may then prepare them for the next data transaction.

An ENDEPIN[0] event will be generated when the data has been transferred from memory to the USBD peripheral.

Finally, an EP0DATADONE event will be generated when the data has been transmitted over USB and acknowledged by the host.

The software can then either prepare and transmit the next data transaction by repeating the above sequence, or initiate the status stage through the EP0STATUS task.

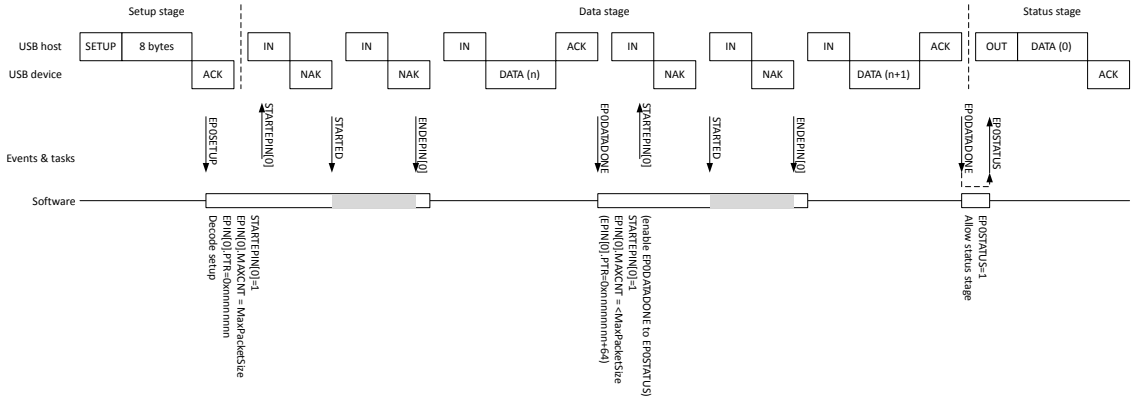

Figure 237: Control read transfer

It is possible to enable a shortcut from the EP0DATADONE event to the EP0STATUS task, typically if the data stage is expected to take a single transfer. If there is no data stage, the software can initiate the status stage through the EP0STATUS task right away, as as shown in the following figure.

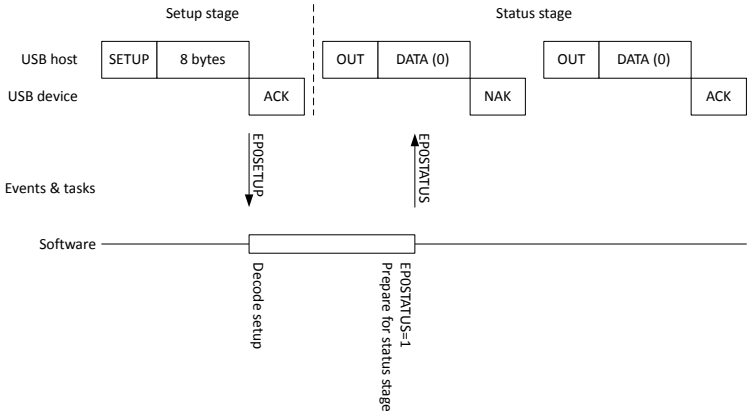

Figure 238: Control read no data transfer

7.39.9.2 Control write transfer

This section describes how the software responds to a control write transfer.

The software has to prepare EasyDMA by pointing to the buffer in memory that shall contain the incoming data. If no other EasyDMA transfers are ongoing with USBD, the software can then send the EP0RCVOUT task, which will make USBD acknowledge (ACK) the first OUT+DATA transaction from the host.

An EP0DATADONE event will be generated when a new OUT+DATA has been transmitted over USB, and is about to get acknowledged by the device.

After receiving the first transaction, a STARTED event (the EPOUT0 bit set in the EPSTATUS register) is generated when the EPOUT[0].PTR and .MAXCNT registers have been captured. Software may then prepare them for the next data transaction.

An ENDEPOUT[0] event will be generated when the data has been transferred from the USBD peripheral to memory. The software can then either prepare to receive the next data transaction by repeating the above sequence, or initiate the status stage through the EP0STATUS task. Until then, further incoming OUT +DATA transactions get a NAK response by the device.

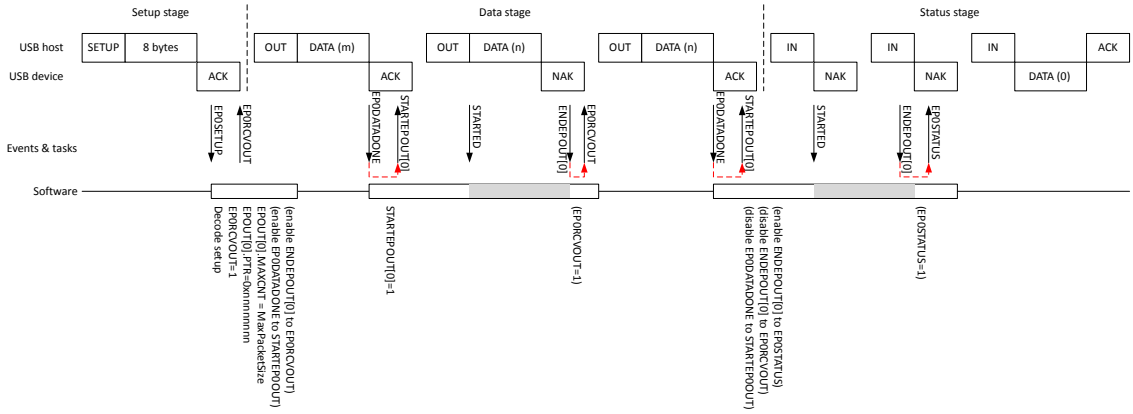

Figure 239: Control write transfer

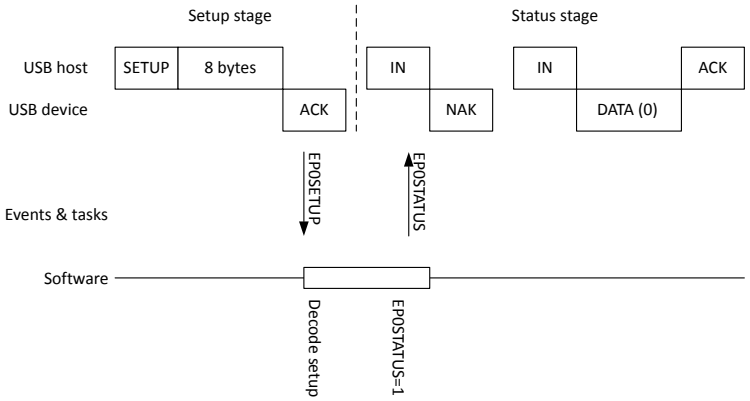

Figure 240: Control write no data transfer

7.39.10 Bulk and interrupt transactions

The USBD peripheral implements seven pairs of bulk/interrupt endpoints.

The bulk/interrupt endpoints have a fixed USB endpoint number, summarized in the following table.

Bulk endpoint #	USB IN endpoint	USB OUT endpoint
[1]	0x81	0x01
[2]	0x82	0x02
[3]	0x83	0x03
[4]	0x84	0x04
[5]	0x85	0x05
[6]	0x86	0x06
[7]	0x87	0x07

Table 177: Bulk/interrupt endpoint numbering

A bulk/interrupt transaction consists of a single data stage. Two consecutive, successful transactions are distinguished through alternating leading process ID (PID): DATA0 follows DATA1, DATA1 follows DATA0, etc. A repeated transaction is detected by re-using the same PID as previous transaction, i.e DATA0 follows DATA0, or DATA1 follows DATA1.

The USBD controller automatically toggles DATA0/DATA1 PIDs for every bulk/interrupt transaction.

If incoming data is corrupted (CRC does not match), the USBD controller automatically prevents DATA0/DATA1 from toggling, to request the host to resend the data.

In some specific cases, the software may want to force a data toggle (usually reset) on a specific IN endpoint, or force the expected toggle on an OUT endpoint, for instance as a consequence of the host issuing **`ClearFeature`**, **`SetInterface`**, or selecting an alternate setting. Controlling the data toggle of data IN or OUT endpoint n (n=1..7) is done through register DTOGGLE.

The bulk/interrupt transaction in USB full-speed can be of any size up to 64 bytes. It must be a multiple of four bytes and 32-bit aligned in memory.

When the USB transaction has completed, an EPDATA event is generated. Until new data has been transferred by EasyDMA from memory to the USBD peripheral (signalled by the ENDEPIN[n] event), the hardware will automatically respond with NAK to all incoming IN tokens. Software has to configure and start the EasyDMA transfer once it is ready to send more data.

Each IN or OUT data endpoint has to be explicitly enabled by software through register EPINEN or EPOUTEN, according to the configuration declared by the device and selected by the host through the **`SetConfig`** command.

A disabled data endpoint will not respond to any traffic from the host. An enabled data endpoint will normally respond NAK or ACK (depending on the readiness of the buffers), or STALL (if configured in register EPSTALL), in which case the endpoint is asked to halt. The halted (or not) state of a given endpoint can be read back from register HALTED.EPIN[n] or HALTED.EPOUT[n]. The format of the returned 16-bit value can be copied as is, as a response to a **`GetStatusEndpoint`** request from the host.

Enabling or disabling an endpoint will not change its halted state. However, a USB reset will disable and clear the halted state of all data endpoints.

The control endpoint 0 IN and OUT can also be enabled and/or halted using the same mechanisms, but due to USB specification, receiving a SETUP will override its state.

7.39.10.1 Bulk and interrupt IN transaction

The host issues IN tokens to receive bulk/interrupt data. In order to send data, the software has to enable the endpoint and prepare an EasyDMA transfer on the desired endpoint.

Bulk/interrupt IN endpoints are enabled or disabled through their respective INn bit (n=1..7) in EPINEN register.

It is also possible to stall or resume communication on an endpoint through the EPSTALL register.

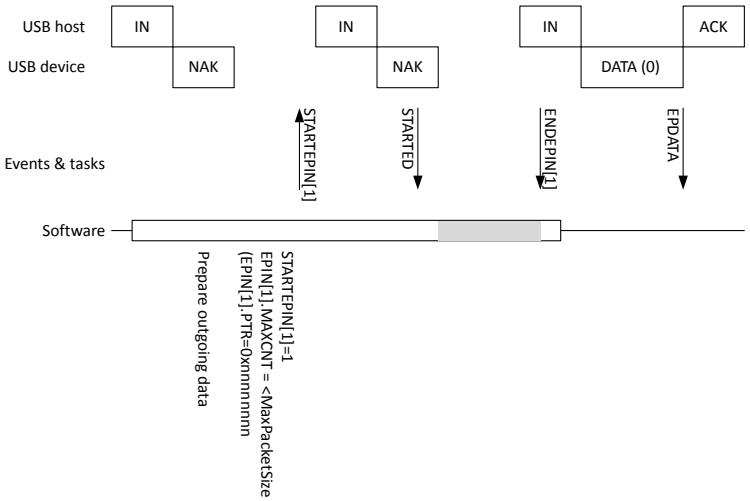

Figure 241: Bulk/interrupt IN transaction

It is possible (and in some situations it is required) to respond to an IN token with a zero-length data packet.

> **Note:** On many USB hosts, not responding (DATA+ACK or NAK) to three IN tokens on an interrupt endpoint would have the host disable that endpoint as a consequence. Re-enumerating the device (unplug-replug) may be required to restore functionality. Make sure that the relevant data endpoints are enabled for normal operation as soon as the device gets configured through a **`SetConfig`** request.

7.39.10.2 Bulk and interrupt OUT transaction

When the host wants to transmit bulk/interrupt data, it issues an OUT token (packet) followed by a DATA packet on a given endpoint n (n=1..7).

A NAK is returned until the software writes any value to register SIZE.EPOUT[n], indicating that the content of the local buffer can be overwritten. Upon receiving the next OUT+DATA transaction, an ACK is returned to the host while an EPDATA event is generated (and the EPDATASTATUS register flags are set to indicate on which endpoint this happened). Once the EasyDMA is prepared and enabled, by writing the EPOUT[n] registers and triggering the STARTEPOUT[n] task, the incoming data will be transferred to memory. Until that transfer is finished, the hardware will automatically NAK any other incoming OUT+DATA packets. Only when the EasyDMA transfer is done (signalled by the ENDEPOUT[n] event), or as soon as any values are written by the software in register SIZE.EPOUT[n], the endpoint n will accept incoming OUT+DATA again.

It is allowed for the host to send zero-length data packets.

Bulk/interrupt OUT endpoints are enabled or disabled through their respective OUTn bit (n=1..7) in the EPOUTEN register. It is also possible to stall or resume communication on an endpoint through the EPSTALL register.

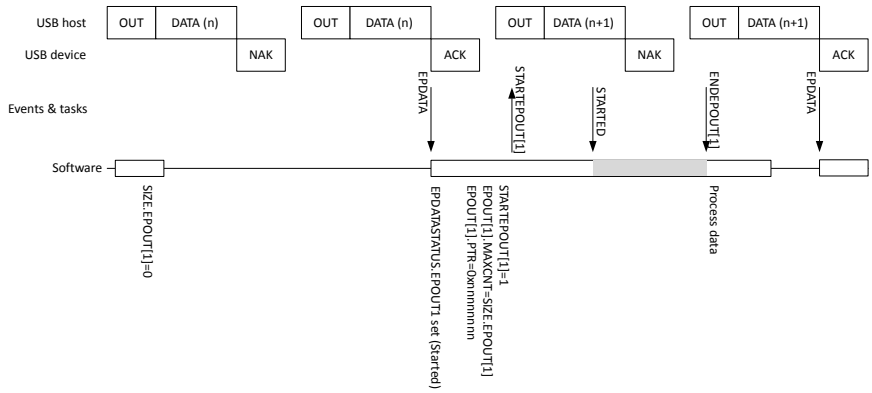

Figure 242: Bulk/interrupt OUT transaction

7.39.11 Isochronous transactions

The USBD peripheral implements isochronous (ISO) endpoints.

The ISO endpoints have a fixed USB endpoint number, summarized in the following table.

ISO endpoint #	USB IN endpoint	USB OUT endpoint
[0]	0x88	0x08

Table 178: Isochronous endpoint numbering

An isochronous transaction consists of a single, non-acknowledged data stage. The host sends out a start of frame at a regular interval (1 ms), and data follows IN or OUT tokens within each frame.

EasyDMA allows transferring ISO data directly from and to memory. EasyDMA transfers must be initiated by the software, which can synchronize with the SOF (start of frame) events.

Because the timing of the start of frame is very accurate, the SOF event can be used for jobs such as synchronizing a local timer through the SOF event and PPI. The SOF event gets synchronized to the 16 MHz clock prior to being made available to the PPI.

Every start of frame increments a free-running counter, which can be read by software through the FRAMECNTR register.

Each IN or OUT ISO data endpoint has to be explicitly enabled by software through register EPINEN or EPOUTEN, according to the configuration declared by the device and selected by the host through the **`SetConfig`** command. A disabled ISO IN data endpoint will not respond to any traffic from the host. A disabled ISO OUT data endpoint will ignore any incoming traffic from the host.

The USBD peripheral has an internal 1 kB buffer associated with ISO endpoints. The user can either allocate the full amount to the IN or the OUT endpoint, or split the buffer allocation between the two using register ISOSPLIT.

The internal buffer also sets the maximum size of the ISO OUT and ISO IN transfers: 1023 bytes when the full buffer is dedicated to either ISO OUT or ISO IN, and half when the buffer is split between the two.

7.39.11.1 Isochronous IN transaction

When the host wants to receive isochronous (ISO) data, it issues an IN token on the isochronous endpoint.

After the data has been transferred using the EasyDMA, the USB controller on the isochronous IN endpoint responds to the IN token with the transferred data using the ISOIN.MAXCNT for the size of the packet.

The ISO IN data endpoint has to be explicitly enabled by software through the ISOIN0 bit in register EPINEN.

When an ISO IN endpoint is enabled and no data transferred with EasyDMA, the response of the USBD depends on the setting of the RESPONSE field in register ISOINCONFIG. It can either provide no response to an IN token or respond with a zero-length data.

If the EasyDMA transfer on the isochronous endpoint is not completed before the next SOF event, the result of the transfer is undefined.

The maximum size of an ISO IN transfer in USB full-speed is 1023 bytes. The data buffer has to be a multiple of 4 bytes 32-bit aligned in memory. However, the amount of bytes transferred on the USB data endpoint can be of any size (up to 1023 bytes, if not shared with an OUT ISO endpoint).

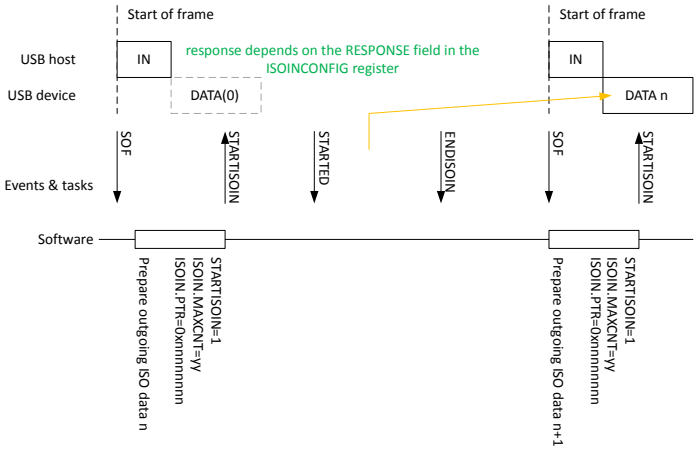

Figure 243: Isochronous IN transfer

7.39.11.2 Isochronous OUT transaction

When the host wants to send isochronous (ISO) data, it issues an OUT token on the isochronous endpoint, followed by data.

The ISO OUT data endpoint has to be explicitly enabled by software through the ISOOUT0 bit in register EPOUTEN.

The amount of last received ISO OUT data is provided in the SIZE.ISOOUT register. Software shall interpret the ZERO and SIZE fields as presented in the following table.

ZERO	SIZE	Last received data size
Normal	0	No data received at all
Normal	1..1023	1..1023 bytes of data received
ZeroData	(not of interest)	Zero-length data packet received

Table 179: ISO OUT incoming data size

When EasyDMA is prepared and started, triggering a STARTISOOUT task initiates an EasyDMA transfer to memory. Software shall synchronize ISO OUT transfers with the SOF events. EasyDMA uses the address in ISOOUT.PTR and size in ISOOUT.MAXCNT for every new transfer.

If the EasyDMA transfer on the isochronous endpoint is not completed before the next SOF event, the result of the transfer is undefined.

The maximum size of an isochronous OUT transfer in USB full-speed is 1023 bytes. The data buffer has to be a multiple of 4 bytes and 32-bit aligned in Data RAM. However, the amount of bytes transferred on the USB data endpoint can be of any size (up to 1023 bytes if not shared with an IN ISO endpoint).

If the last received ISO data packet is corrupted (wrong CRC), the USB controller generates an USBEVENT event (at the same time as SOF) and indicates a CRC error on ISOOUTCRC in register EVENTCAUSE. EasyDMA will transfer the data anyway if it has been set up properly.

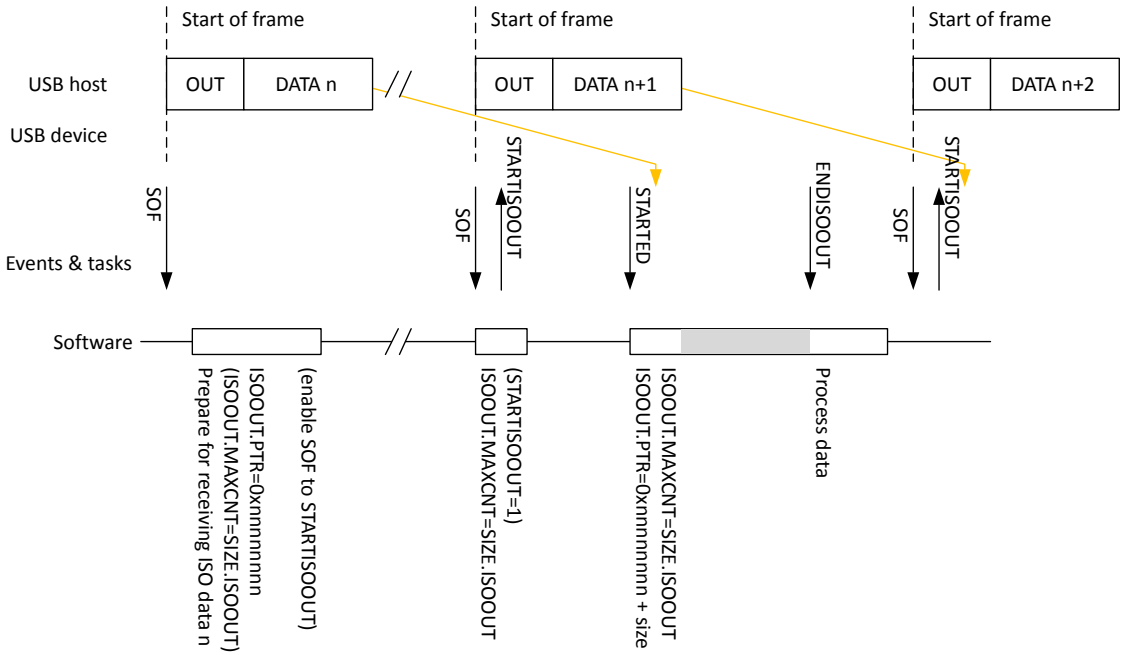

Figure 244: Isochronous OUT transfer

7.39.12 USB register access limitations

Some of the registers in USBD cannot be accessed in specific conditions.

This may be the case when USBD is not enabled (using the ENABLE register) and ready (signalled by the READY bit in EVENTCAUSE after a USBEVENT event), or when USBD is in low power mode while the USB bus is suspended.

Triggering any tasks, including the tasks triggered through the PPI, is affected by this behavior. In addition, the following registers are affected:

- HALTED.EPIN[0..7]
- HALTED.EPOUT[0..7]
- USBADDR
- BMREQUESTTYPE
- BREQUEST
- WVALUEL
- WVALUEH
- WINDEXL
- WINDEXH
- WLENGTHL
- WLENGTHH
- SIZE.EPOUT[0..7]
- SIZE.ISOOUT
- USBPULLUP
- DTOGGLE
- EPINEN
- EPOUTEN
- EPSTALL
- ISOSPLIT
- FRAMECNTR

7.39.13 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50036000 0x40036000	APPLICATION	USBD	USBD : S USBD : NS	US	SA	Universal serial bus device	

Table 180: Instances

Register	Offset	Security	Description
TASKS_STARTEPIN[n]	0x004		Captures the EPIN[n].PTR and EPIN[n].MAXCNT registers values, and enables endpoint IN n to respond to traffic from host
TASKS_STARTISOIN	0x024		Captures the ISOIN.PTR and ISOIN.MAXCNT registers values, and enables sending data on ISO endpoint
TASKS_STARTEPOUT[n]	0x028		Captures the EPOUT[n].PTR and EPOUT[n].MAXCNT registers values, and enables endpoint n to respond to traffic from host
TASKS_STARTISOOUT	0x048		Captures the ISOOUT.PTR and ISOOUT.MAXCNT registers values, and enables receiving of data on ISO endpoint
TASKS_EP0RCVOUT	0x04C		Allows OUT data stage on control endpoint 0
TASKS_EP0STATUS	0x050		Allows status stage on control endpoint 0
TASKS_EP0STALL	0x054		Stalls data and status stage on control endpoint 0
TASKS_DPDMDRIVE	0x058		Forces D+ and D- lines into the state defined in the DPDMVALUE register

Register	Offset	Security	Description
TASKS_DPDMNODRIVE	0x05C		Stops forcing D+ and D- lines into any state (USB engine takes control)
SUBSCRIBE_STARTEPIN[n]	0x084		Subscribe configuration for task STARTEPIN[n]
SUBSCRIBE_STARTISOIN	0x0A4		Subscribe configuration for task STARTISOIN
SUBSCRIBE_STARTEPOUT[n]	0x0A8		Subscribe configuration for task STARTEPOUT[n]
SUBSCRIBE_STARTISOOUT	0x0C8		Subscribe configuration for task STARTISOOUT
SUBSCRIBE_EP0RCVOUT	0x0CC		Subscribe configuration for task EP0RCVOUT
SUBSCRIBE_EP0STATUS	0x0D0		Subscribe configuration for task EP0STATUS
SUBSCRIBE_EP0STALL	0x0D4		Subscribe configuration for task EP0STALL
SUBSCRIBE_DPDMDRIVE	0x0D8		Subscribe configuration for task DPDMDRIVE
SUBSCRIBE_DPDMNODRIVE	0x0DC		Subscribe configuration for task DPDMNODRIVE
EVENTS_USBRESET	0x100		Signals that a USB reset condition has been detected on USB lines
EVENTS_STARTED	0x104		Confirms that the EPIN[n].PTR and EPIN[n].MAXCNT, or EPOUT[n].PTR and EPOUT[n].MAXCNT registers have been captured on all endpoints reported in the EPSTATUS register
EVENTS_ENDEPIN[n]	0x108		The whole EPIN[n] buffer has been consumed. The buffer can be accessed safely by software.
EVENTS_EP0DATADONE	0x128		An acknowledged data transfer has taken place on the control endpoint
EVENTS_ENDISOIN	0x12C		The whole ISOIN buffer has been consumed. The buffer can be accessed safely by software.
EVENTS_ENDEPOUT[n]	0x130		The whole EPOUT[n] buffer has been consumed. The buffer can be accessed safely by software.
EVENTS_ENDISOOUT	0x150		The whole ISOOUT buffer has been consumed. The buffer can be accessed safely by software.
EVENTS_SOF	0x154		Signals that a SOF (start of frame) condition has been detected on USB lines
EVENTS_USBEVENT	0x158		An event or an error not covered by specific events has occurred. Check EVENTCAUSE register to find the cause.
EVENTS_EP0SETUP	0x15C		A valid SETUP token has been received (and acknowledged) on the control endpoint
EVENTS_EPDATA	0x160		A data transfer has occurred on a data endpoint, indicated by the EPDATASTATUS register
PUBLISH_USBRESET	0x180		Publish configuration for event USBRESET
PUBLISH_STARTED	0x184		Publish configuration for event STARTED
PUBLISH_ENDEPIN[n]	0x188		Publish configuration for event ENDEPIN[n]
PUBLISH_EP0DATADONE	0x1A8		Publish configuration for event EP0DATADONE
PUBLISH_ENDISOIN	0x1AC		Publish configuration for event ENDISOIN
PUBLISH_ENDEPOUT[n]	0x1B0		Publish configuration for event ENDEPOUT[n]
PUBLISH_ENDISOOUT	0x1D0		Publish configuration for event ENDISOOUT
PUBLISH_SOF	0x1D4		Publish configuration for event SOF
PUBLISH_USBEVENT	0x1D8		Publish configuration for event USBEVENT
PUBLISH_EP0SETUP	0x1DC		Publish configuration for event EP0SETUP
PUBLISH_EPDATA	0x1E0		Publish configuration for event EPDATA
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
EVENTCAUSE	0x400		Details on what caused the USBEVENT event
HALTED.EPIN[n]	0x420		IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
HALTED.EPOUT[n]	0x444		OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
EPSTATUS	0x468		Provides information on which endpoint's EasyDMA registers have been captured
EPDATASTATUS	0x46C		Provides information on which endpoint(s) an acknowledged data transfer has occurred (EPDATA event)
USBADDR	0x470		Device USB address

Register	Offset	Security	Description
BMREQUESTTYPE	0x480		SETUP data, byte 0, bmRequestType
BREQUEST	0x484		SETUP data, byte 1, bRequest
WVALUEL	0x488		SETUP data, byte 2, LSB of wValue
WVALUEH	0x48C		SETUP data, byte 3, MSB of wValue
WINDEXL	0x490		SETUP data, byte 4, LSB of wIndex
WINDEXH	0x494		SETUP data, byte 5, MSB of wIndex
WLENGTHL	0x498		SETUP data, byte 6, LSB of wLength
WLENGTHH	0x49C		SETUP data, byte 7, MSB of wLength
SIZE.EPOUT[n]	0x4A0		Number of bytes received last in the data stage of this OUT endpoint
SIZE.ISOOUT	0x4C0		Number of bytes received last on this ISO OUT data endpoint
ENABLE	0x500		Enable USB
USBPULLUP	0x504		Control of the USB pull-up
DPDMVALUE	0x508		State D+ and D- lines will be forced into by the DPDMDRIVE task. The DPDMNODRIVE task reverts the control of the lines to MAC IP (no forcing).
DTOGGLE	0x50C		Data toggle control and status
EPINEN	0x510		Endpoint IN enable
EPOUTEN	0x514		Endpoint OUT enable
EPSTALL	0x518		STALL endpoints
ISOSPLIT	0x51C		Controls the split of ISO buffers
FRAMECNTR	0x520		Returns the current value of the start of frame counter
LOWPOWER	0x52C		Controls USBD peripheral low power mode during USB suspend
ISOINCONFIG	0x530		Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent
EPIN[n].PTR	0x600		Data pointer
EPIN[n].MAXCNT	0x604		Maximum number of bytes to transfer
EPIN[n].AMOUNT	0x608		Number of bytes transferred in the last transaction
ISOIN.PTR	0x6A0		Data pointer
ISOIN.MAXCNT	0x6A4		Maximum number of bytes to transfer
ISOIN.AMOUNT	0x6A8		Number of bytes transferred in the last transaction
EPOUT[n].PTR	0x700		Data pointer
EPOUT[n].MAXCNT	0x704		Maximum number of bytes to transfer
EPOUT[n].AMOUNT	0x708		Number of bytes transferred in the last transaction
ISOOUT.PTR	0x7A0		Data pointer
ISOOUT.MAXCNT	0x7A4		Maximum number of bytes to transfer
ISOOUT.AMOUNT	0x7A8		Number of bytes transferred in the last transaction

Table 181: Register overview

7.39.13.1 TASKS_STARTEPIN[n] (n=0..7)

Address offset: 0x004 + (n × 0x4)

Captures the EPIN[n].PTR and EPIN[n].MAXCNT registers values, and enables endpoint IN n to respond to traffic from host

Bit number	31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTEPIN			Captures the EPIN[n].PTR and EPIN[n].MAXCNT registers values, and enables endpoint IN n to respond to traffic from host
			Trigger	1	Trigger task

7.39.13.2 TASKS_STARTISOIN

Address offset: 0x024

Captures the ISOIN.PTR and ISOIN.MAXCNT registers values, and enables sending data on ISO endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STARTISOIN			Captures the ISOIN.PTR and ISOIN.MAXCNT registers values, and enables sending data on ISO endpoint
			Trigger	1	Trigger task

7.39.13.3 TASKS_STARTEPOUT[n] (n=0..7)

Address offset: 0x028 + (n × 0x4)

Captures the EPOUT[n].PTR and EPOUT[n].MAXCNT registers values, and enables endpoint n to respond to traffic from host

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STARTEPOUT			Captures the EPOUT[n].PTR and EPOUT[n].MAXCNT registers values, and enables endpoint n to respond to traffic from host
			Trigger	1	Trigger task

7.39.13.4 TASKS_STARTISOOUT

Address offset: 0x048

Captures the ISOOUT.PTR and ISOOUT.MAXCNT registers values, and enables receiving of data on ISO endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_STARTISOOUT			Captures the ISOOUT.PTR and ISOOUT.MAXCNT registers values, and enables receiving of data on ISO endpoint
			Trigger	1	Trigger task

7.39.13.5 TASKS_EP0RCVOUT

Address offset: 0x04C

Allows OUT data stage on control endpoint 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_EP0RCVOUT			Allows OUT data stage on control endpoint 0
			Trigger	1	Trigger task

7.39.13.6 TASKS_EP0STATUS

Address offset: 0x050

Allows status stage on control endpoint 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_EP0STATUS			Allows status stage on control endpoint 0
			Trigger	1	Trigger task

7.39.13.7 TASKS_EP0STALL

Address offset: 0x054

Stalls data and status stage on control endpoint 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_EP0STALL			Stalls data and status stage on control endpoint 0
			Trigger	1	Trigger task

7.39.13.8 TASKS_DPDMDRIVE

Address offset: 0x058

Forces D+ and D- lines into the state defined in the DPDMVALUE register

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

ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_DPDMDRIVE			Forces D+ and D- lines into the state defined in the DPDMVALUE register
			Trigger	1	Trigger task

7.39.13.9 TASKS_DPDMNODRIVE

Address offset: 0x05C

Stops forcing D+ and D- lines into any state (USB engine takes control)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	TASKS_DPDMNODRIVE			Stops forcing D+ and D- lines into any state (USB engine takes control)
			Trigger	1	Trigger task

7.39.13.10 SUBSCRIBE_STARTEPIN[n] (n=0..7)

Address offset: 0x084 + (n × 0x4)

Subscribe configuration for task STARTEPIN[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTEPIN[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.11 SUBSCRIBE_STARTISOIN

Address offset: 0x0A4

Subscribe configuration for task STARTISOIN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTISOIN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.12 SUBSCRIBE_STARTEPOUT[n] (n=0..7)

Address offset: 0x0A8 + (n × 0x4)

Subscribe configuration for task STARTEPOUT[n]

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTEPOUT[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.13 SUBSCRIBE_STARTISOOUT

Address offset: 0x0C8

Subscribe configuration for task STARTISOOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STARTISOOUT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.14 SUBSCRIBE_EP0RCVOUT

Address offset: 0x0CC

Subscribe configuration for task EP0RCVOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task EP0RCVOUT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.15 SUBSCRIBE_EP0STATUS

Address offset: 0x0D0

Subscribe configuration for task EP0STATUS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task EP0STATUS will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.16 SUBSCRIBE_EP0STALL

Address offset: 0x0D4

Subscribe configuration for task EP0STALL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task EP0STALL will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.17 SUBSCRIBE_DPDMDRIVE

Address offset: 0x0D8

Subscribe configuration for task DPDMDRIVE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task DPDMDRIVE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.18 SUBSCRIBE_DPDMNODRIVE

Address offset: 0x0DC

Subscribe configuration for task DPDMNODRIVE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task DPDMNODRIVE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.39.13.19 EVENTS_USBRESET

Address offset: 0x100

Signals that a USB reset condition has been detected on USB lines

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_USBRESET			Signals that a USB reset condition has been detected on USB lines
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.20 EVENTS_STARTED

Address offset: 0x104

Confirms that the EPIN[n].PTR and EPIN[n].MAXCNT, or EPOUT[n].PTR and EPOUT[n].MAXCNT registers have been captured on all endpoints reported in the EPSTATUS register

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			Confirms that the EPIN[n].PTR and EPIN[n].MAXCNT, or EPOUT[n].PTR and EPOUT[n].MAXCNT registers have been captured on all endpoints reported in the EPSTATUS register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.21 EVENTS_ENDEPIN[n] (n=0..7)

Address offset: 0x108 + (n × 0x4)

The whole EPIN[n] buffer has been consumed. The buffer can be accessed safely by software.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDEPIN			The whole EPIN[n] buffer has been consumed. The buffer can be accessed safely by software.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.22 EVENTS_EP0DATADONE

Address offset: 0x128

An acknowledged data transfer has taken place on the control endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_EP0DATADONE			An acknowledged data transfer has taken place on the control endpoint
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.23 EVENTS_ENDISOIN

Address offset: 0x12C

The whole ISOIN buffer has been consumed. The buffer can be accessed safely by software.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDISOIN			The whole ISOIN buffer has been consumed. The buffer can be accessed safely by software.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.24 EVENTS_ENDEPOUT[n] (n=0..7)

Address offset: 0x130 + (n × 0x4)

The whole EPOUT[n] buffer has been consumed. The buffer can be accessed safely by software.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDEPOUT			The whole EPOUT[n] buffer has been consumed. The buffer can be accessed safely by software.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.25 EVENTS_ENDISOOUT

Address offset: 0x150

The whole ISOOUT buffer has been consumed. The buffer can be accessed safely by software.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_ENDISOOUT			The whole ISOOUT buffer has been consumed. The buffer can be accessed safely by software.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.26 EVENTS_SOF

Address offset: 0x154

Signals that a SOF (start of frame) condition has been detected on USB lines

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_SOF			Signals that a SOF (start of frame) condition has been detected on USB lines
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.27 EVENTS_USBEVENT

Address offset: 0x158

An event or an error not covered by specific events has occurred. Check EVENTCAUSE register to find the cause.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_USBEVENT			An event or an error not covered by specific events has occurred. Check EVENTCAUSE register to find the cause.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.28 EVENTS_EP0SETUP

Address offset: 0x15C

A valid SETUP token has been received (and acknowledged) on the control endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_EP0SETUP			A valid SETUP token has been received (and acknowledged) on the control endpoint
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.29 EVENTS_EPDATA

Address offset: 0x160

A data transfer has occurred on a data endpoint, indicated by the EPDATASTATUS register

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

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_EPDATA			A data transfer has occurred on a data endpoint, indicated by the EPDATASTATUS register
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.39.13.30 PUBLISH_USBRESET

Address offset: 0x180

Publish configuration for event USBRESET

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event USBRESET will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.31 PUBLISH_STARTED

Address offset: 0x184

Publish configuration for event STARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.32 PUBLISH_ENDEPIN[n] (n=0..7)

Address offset: 0x188 + (n × 0x4)

Publish configuration for event ENDEPIN[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDEPIN[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.33 PUBLISH_EP0DATADONE

Address offset: 0x1A8

Publish configuration for event EP0DATADONE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event EP0DATADONE will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.34 PUBLISH_ENDISOIN

Address offset: 0x1AC

Publish configuration for event ENDISOIN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDISOIN will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.35 PUBLISH_ENDEPOUT[n] (n=0..7)

Address offset: 0x1B0 + (n × 0x4)

Publish configuration for event ENDEPOUT[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDEPOUT[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.36 PUBLISH_ENDISOOUT

Address offset: 0x1D0

Publish configuration for event ENDISOOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event ENDISOOUT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.37 PUBLISH_SOF

Address offset: 0x1D4

Publish configuration for event SOF

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event SOF will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.38 PUBLISH_USBEVENT

Address offset: 0x1D8

Publish configuration for event USBEVENT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event USBEVENT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.39 PUBLISH_EP0SETUP

Address offset: 0x1DC

Publish configuration for event EP0SETUP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event EP0SETUP will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.40 PUBLISH_EPDATA

Address offset: 0x1E0

Publish configuration for event EPDATA

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event EPDATA will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.39.13.41 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EP0DATADONE_STARTEPIN0			Shortcut between event EP0DATADONE and task STARTEPIN[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	EP0DATADONE_STARTEP			Shortcut between event EP0DATADONE and task STARTEPOUT[0]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	EP0DATADONE_EP0STATUS			Shortcut between event EP0DATADONE and task EP0STATUS
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDEPOUT0_EP0STATUS			Shortcut between event ENDEPOUT[0] and task EP0STATUS
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	ENDEPOUT0_EP0RCVOUT			Shortcut between event ENDEPOUT[0] and task EP0RCVOUT
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

7.39.13.42 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	USBRESET			Enable or disable interrupt for event USBRESET
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	STARTED			Enable or disable interrupt for event STARTED
			Disabled	0	Disable
			Enabled	1	Enable
C-J	RW	ENDEPIN[i] (i=0..7)			Enable or disable interrupt for event ENDEPIN[i]
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	EP0DATADONE			Enable or disable interrupt for event EP0DATADONE
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	ENDISOIN			Enable or disable interrupt for event ENDISOIN
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
M-T	RW	ENDEPOUT[i] (i=0..7)			Enable or disable interrupt for event ENDEPOUT[i]
			Disabled	0	Disable
			Enabled	1	Enable
U	RW	ENDISOOUT			Enable or disable interrupt for event ENDISOOUT
			Disabled	0	Disable
			Enabled	1	Enable
V	RW	SOF			Enable or disable interrupt for event SOF
			Disabled	0	Disable
			Enabled	1	Enable
W	RW	USBEVENT			Enable or disable interrupt for event USBEVENT
			Disabled	0	Disable
			Enabled	1	Enable
X	RW	EP0SETUP			Enable or disable interrupt for event EP0SETUP
			Disabled	0	Disable
			Enabled	1	Enable
Y	RW	EPDATA			Enable or disable interrupt for event EPDATA
			Disabled	0	Disable
			Enabled	1	Enable

7.39.13.43 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	USBRESET			Write '1' to enable interrupt for event USBRESET
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C-J	RW	ENDEPIN[i] (i=0..7)			Write '1' to enable interrupt for event ENDEPIN[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	EP0DATADONE			Write '1' to enable interrupt for event EP0DATADONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	ENDISOIN			Write '1' to enable interrupt for event ENDISOIN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M-T	RW	ENDEPOUT[i] (i=0..7)			Write '1' to enable interrupt for event ENDEPOUT[i]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	ENDISOOUT			Write '1' to enable interrupt for event ENDISOOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	SOF			Write '1' to enable interrupt for event SOF
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
W	RW	USBEVENT			Write '1' to enable interrupt for event USBEVENT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
X	RW	EP0SETUP			Write '1' to enable interrupt for event EP0SETUP
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Y	RW	EPDATA			Write '1' to enable interrupt for event EPDATA
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.39.13.44 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	USBRESET			Write '1' to disable interrupt for event USBRESET
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C-J	RW	ENDEPIN[i] (i=0..7)			Write '1' to disable interrupt for event ENDEPIN[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	EP0DATADONE			Write '1' to disable interrupt for event EP0DATADONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				Y X W V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
L	RW	ENDISOIN			Write '1' to disable interrupt for event ENDISOIN
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M-T	RW	ENDEPOUT[i] (i=0..7)			Write '1' to disable interrupt for event ENDEPOUT[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
U	RW	ENDISOOUT			Write '1' to disable interrupt for event ENDISOOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	SOF			Write '1' to disable interrupt for event SOF
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
W	RW	USBEVENT			Write '1' to disable interrupt for event USBEVENT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
X	RW	EP0SETUP			Write '1' to disable interrupt for event EP0SETUP
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Y	RW	EPDATA			Write '1' to disable interrupt for event EPDATA
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.39.13.45 EVENTCAUSE

Address offset: 0x400

Details on what caused the USBEVENT event

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ISOOUTCRC			CRC error was detected on isochronous OUT endpoint 8. Write '1' to clear.
			NotDetected	0	No error detected
			Detected	1	Error detected
B	RW	SUSPEND			Signals that USB lines have been idle long enough for the device to enter suspend. Write '1' to clear.
			NotDetected	0	Suspend not detected
			Detected	1	Suspend detected
C	RW	RESUME			Signals that a RESUME condition (K state or activity restart) has been detected on USB lines. Write '1' to clear.
			NotDetected	0	Resume not detected

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

ID	R/W	Field	Value ID	Value	Description
			Detected	1	Resume detected
D	RW	USBWUALLOWED			USB MAC has been woken up and operational. Write '1' to clear.
			NotAllowed	0	Wake up not allowed
			Allowed	1	Wake up allowed
E	RW	READY			USB device is ready for normal operation. Write '1' to clear.
			NotDetected	0	USBEVENT was not issued due to USBD peripheral ready
			Ready	1	USBD peripheral is ready

7.39.13.46 HALTED.EPIN[n] (n=0..7)

Address offset: 0x420 + (n × 0x4)

IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.

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

ID	R/W	Field	Value ID	Value	Description
A	R	GETSTATUS			IN endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
			NotHalted	0	Endpoint is not halted
			Halted	1	Endpoint is halted

7.39.13.47 HALTED.EPOUT[n] (n=0..7)

Address offset: 0x444 + (n × 0x4)

OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.

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

ID	R/W	Field	Value ID	Value	Description
A	R	GETSTATUS			OUT endpoint halted status. Can be used as is as response to a GetStatus() request to endpoint.
			NotHalted	0	Endpoint is not halted
			Halted	1	Endpoint is halted

7.39.13.48 EPSTATUS

Address offset: 0x468

Provides information on which endpoint's EasyDMA registers have been captured

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-I	RW	EPIN[i] (i=0..8)			Captured state of endpoint's EasyDMA registers. Write '1' to clear.
			NoData	0	EasyDMA registers have not been captured for this endpoint
			DataDone	1	EasyDMA registers have been captured for this endpoint
J-R	RW	EPOUT[i] (i=0..8)			Captured state of endpoint's EasyDMA registers. Write '1' to clear.
			NoData	0	EasyDMA registers have not been captured for this endpoint
			DataDone	1	EasyDMA registers have been captured for this endpoint

7.39.13.49 EPDATASTATUS

Address offset: 0x46C

Provides information on which endpoint(s) an acknowledged data transfer has occurred (EPDATA event)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-G	RW	EPIN[i] (i=1..7)			Acknowledged data transfer on this IN endpoint. Write '1' to clear.
			NotDone	0	No acknowledged data transfer on this endpoint
			DataDone	1	Acknowledged data transfer on this endpoint has occurred
H-N	RW	EPOUT[i] (i=1..7)			Acknowledged data transfer on this OUT endpoint. Write '1' to clear.
			NotStarted	0	No acknowledged data transfer on this endpoint
			Started	1	Acknowledged data transfer on this endpoint has occurred

7.39.13.50 USBADDR

Address offset: 0x470

Device USB address

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	ADDR			Device USB address

7.39.13.51 BMREQUESTTYPE

Address offset: 0x480

SETUP data, byte 0, bmRequestType

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B B A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	RECIPIENT			Data transfer type
			Device	0	Device
			Interface	1	Interface
			Endpoint	2	Endpoint
			Other	3	Other
B	R	TYPE			Data transfer type
			Standard	0	Standard
			Class	1	Class
			Vendor	2	Vendor
C	R	DIRECTION			Data transfer direction
			HostToDevice	0	Host-to-device
			DeviceToHost	1	Device-to-host

7.39.13.52 BREQUEST

Address offset: 0x484

SETUP data, byte 1, bRequest

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	BREQUEST			SETUP data, byte 1, bRequest. Values provided for standard requests only, user must implement class and vendor values.
			STD_GET_STATUS	0	Standard request GET_STATUS
			STD_CLEAR_FEATURE	1	Standard request CLEAR_FEATURE
			STD_SET_FEATURE	3	Standard request SET_FEATURE
			STD_SET_ADDRESS	5	Standard request SET_ADDRESS
			STD_GET_DESCRIPTOR	6	Standard request GET_DESCRIPTOR
			STD_SET_DESCRIPTOR	7	Standard request SET_DESCRIPTOR
			STD_GET_CONFIGURATION	8	Standard request GET_CONFIGURATION
			STD_SET_CONFIGURATION	9	Standard request SET_CONFIGURATION
			STD_GET_INTERFACE	10	Standard request GET_INTERFACE
			STD_SET_INTERFACE	11	Standard request SET_INTERFACE
			STD_SYNCH_FRAME	12	Standard request SYNCH_FRAME

7.39.13.53 WVALUEL

Address offset: 0x488

SETUP data, byte 2, LSB of wValue

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	WVALUEL			SETUP data, byte 2, LSB of wValue

7.39.13.54 WVALUEH

Address offset: 0x48C

SETUP data, byte 3, MSB of wValue

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												A	A	A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
A	R	WVALUEH										SETUP data, byte 3, MSB of wValue																							

7.39.13.55 WINDEXL

Address offset: 0x490

SETUP data, byte 4, LSB of wIndex

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												A	A	A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
A	R	WINDEXL										SETUP data, byte 4, LSB of wIndex																							

7.39.13.56 WINDEXH

Address offset: 0x494

SETUP data, byte 5, MSB of wIndex

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												A	A	A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
A	R	WINDEXH										SETUP data, byte 5, MSB of wIndex																							

7.39.13.57 WLENGTHL

Address offset: 0x498

SETUP data, byte 6, LSB of wLength

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												A	A	A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value								Description																							
A	R	WLENGTHL										SETUP data, byte 6, LSB of wLength																							

7.39.13.58 WLENGTHH

Address offset: 0x49C

SETUP data, byte 7, MSB of wLength

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	WLENGTHH			SETUP data, byte 7, MSB of wLength

7.39.13.59 SIZE.EPOUT[n] (n=0..7)

Address offset: 0x4A0 + (n × 0x4)

Number of bytes received last in the data stage of this OUT endpoint

Write to any value to accept further OUT traffic on this endpoint, and overwrite the intermediate buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	SIZE			Number of bytes received last in the data stage of this OUT endpoint

7.39.13.60 SIZE.ISOOUT

Address offset: 0x4C0

Number of bytes received last on this ISO OUT data endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A A
Reset 0x00010000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	SIZE			Number of bytes received last on this ISO OUT data endpoint
B	R	ZERO			Zero-length data packet received
			Normal	0	No zero-length data received, use value in SIZE
			ZeroData	1	Zero-length data received, ignore value in SIZE

7.39.13.61 ENABLE

Address offset: 0x500

Enable USB

After writing Disabled to this register, reading the register will return Enabled until USBD is completely 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ENABLE			Enable USB
			Disabled	0	USB peripheral is disabled
			Enabled	1	USB peripheral is enabled

7.39.13.62 USBPULLUP

Address offset: 0x504

Control of the USB pull-up

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CONNECT			Control of the USB pull-up on the D+ line
			Disabled	0	Pull-up is disconnected
			Enabled	1	Pull-up is connected to D+

7.39.13.63 DPDMVALUE

Address offset: 0x508

State D+ and D- lines will be forced into by the DPDMDRIVE task. The DPDMNODRIVE task reverts the control of the lines to MAC IP (no forcing).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	STATE			State D+ and D- lines will be forced into by the DPDMDRIVE task
			Resume	1	D+ forced low, D- forced high (K state) for a timing preset in hardware (50 µs or 5 ms, depending on bus state)
			J	2	D+ forced high, D- forced low (J state)
			K	4	D+ forced low, D- forced high (K state)

7.39.13.64 DTOGGLE

Address offset: 0x50C

Data toggle control and status

First write this register with VALUE=Nop to select the endpoint, then either read it to get the status from VALUE, or write it again with VALUE=Data0 or Data1

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C C B A A A
Reset 0x00000100				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	EP			Select bulk endpoint number
B	RW	IO			Selects IN or OUT endpoint
			Out	0	Selects OUT endpoint
			In	1	Selects IN endpoint
C	RW	VALUE			Data toggle value
			Nop	0	No action on data toggle when writing the register with this value
			Data0	1	Data toggle is DATA0 on endpoint set by EP and IO
			Data1	2	Data toggle is DATA1 on endpoint set by EP and IO

7.39.13.65 EPINEN

Address offset: 0x510

Endpoint IN enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	IN[i] (i=0..7)			Enable IN endpoint i
			Disable	0	Disable endpoint IN i (no response to IN tokens)
			Enable	1	Enable endpoint IN i (response to IN tokens)
I	RW	ISOIN			Enable ISO IN endpoint
			Disable	0	Disable ISO IN endpoint 8
			Enable	1	Enable ISO IN endpoint 8

7.39.13.66 EPOUTEN

Address offset: 0x514

Endpoint OUT enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				I H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	OUT[i] (i=0..7)			Enable OUT endpoint i
			Disable	0	Disable endpoint OUT i (no response to OUT tokens)
			Enable	1	Enable endpoint OUT i (response to OUT tokens)
I	RW	ISOOUT			Enable ISO OUT endpoint 8
			Disable	0	Disable ISO OUT endpoint 8
			Enable	1	Enable ISO OUT endpoint 8

7.39.13.67 EPSTALL

Address offset: 0x518

STALL endpoints

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

ID	R/W	Field	Value ID	Value	Description
A	W	EP			Select endpoint number
B	W	IO			Selects IN or OUT endpoint
			Out	0	Selects OUT endpoint
			In	1	Selects IN endpoint
C	W	STALL			Stall selected endpoint
			UnStall	0	Don't stall selected endpoint
			Stall	1	Stall selected endpoint

7.39.13.68 ISOSPLIT

Address offset: 0x51C

Controls the split of ISO buffers

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

ID	R/W	Field	Value ID	Value	Description
A	RW	SPLIT			Controls the split of ISO buffers
			OneDir	0x0000	Full buffer dedicated to either ISO IN or OUT
			HalfIN	0x0080	Lower half for IN, upper half for OUT

7.39.13.69 FRAMECNTR

Address offset: 0x520

Returns the current value of the start of frame counter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	FRAMECNTR			Returns the current value of the start of frame counter

7.39.13.70 LOWPOWER

Address offset: 0x52C

Controls USBD peripheral low power mode during USB suspend

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LOWPOWER			Controls USBD peripheral low-power mode during USB suspend
			ForceNormal	0	Software must write this value to exit low power mode and before performing a remote wake-up
			LowPower	1	Software must write this value to enter low power mode after DMA and software have finished interacting with the USB peripheral

7.39.13.71 ISOINCONFIG

Address offset: 0x530

Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RESPONSE			Controls the response of the ISO IN endpoint to an IN token when no data is ready to be sent
			NoResp	0	Endpoint does not respond in that case
			ZeroData	1	Endpoint responds with a zero-length data packet in that case

7.39.13.72 EPIN[n].PTR (n=0..7)

Address offset: 0x600 + (n × 0x14)

Data pointer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.39.13.73 EPIN[n].MAXCNT (n=0..7)

Address offset: 0x604 + (n × 0x14)

Maximum number of bytes to transfer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[64..0]	Maximum number of bytes to transfer

7.39.13.74 EPIN[n].AMOUNT (n=0..7)

Address offset: 0x608 + (n × 0x14)

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

7.39.13.75 ISOIN.PTR

Address offset: 0x6A0

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.39.13.76 ISOIN.MAXCNT

Address offset: 0x6A4

Maximum number of bytes to transfer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1023..1]	Maximum number of bytes to transfer

7.39.13.77 ISOIN.AMOUNT

Address offset: 0x6A8

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

7.39.13.78 EPOUT[n].PTR (n=0..7)

Address offset: 0x700 + (n × 0x14)

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.39.13.79 EPOUT[n].MAXCNT (n=0..7)

Address offset: 0x704 + (n × 0x14)

Maximum number of bytes to transfer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[64..0]	Maximum number of bytes to transfer

7.39.13.80 EPOUT[n].AMOUNT (n=0..7)

Address offset: 0x708 + (n × 0x14)

Number of bytes transferred in the last transaction

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

ID	R/W	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

7.39.13.81 ISOOUT.PTR

Address offset: 0x7A0

Data pointer

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

ID	R/W	Field	Value ID	Value	Description
A	RW	PTR			Data pointer See the memory chapter for details about which memories are available for EasyDMA.

7.39.13.82 ISOOUT.MAXCNT

Address offset: 0x7A4

Maximum number of bytes to transfer

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

7.39.13.83 ISOOUT.AMOUNT

Address offset: 0x7A8

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	AMOUNT			Number of bytes transferred in the last transaction

7.39.14 Electrical specification

7.39.14.1 USB Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$R_{USB,PU,ACTIVE}$	Value of pull-up on D+, bus active (upstream device transmitting)	1425	2300	3090	Ω
$R_{USB,PU,IDLE}$	Value of pull-up on D+, bus idle	900	1200	1575	Ω
$t_{USB,DETRST}$	Minimum duration of an SE0 state to be detected as a USB reset condition			10	µs
$f_{USB,CLK}$	Frequency of local clock, USB active		48		MHz
$f_{USB,TOL}$	Accuracy of local clock, USB active[31]			±1000	ppm
$T_{USB,JITTER}$	Jitter on USB local clock, USB active			±1	ns

7.40 VMC — Volatile memory controller

VMC provides power control for RAM blocks.

Each RAM block, which may contain multiple RAM sections, can power up or power down independently in System ON and System OFF mode using RAM[n] registers. See the Memory chapter for more information about RAM blocks and sections.

7.40.1 RAM power states

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

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

In System OFF, retention of a RAM section is configured in the RETENTION field of the corresponding register RAM[n].POWER (n=0..7) on page 735.

In System ON, retention and accessibility for a RAM section is configured in the RETENTION and POWER fields of the corresponding register RAM[n].POWER (n=0..7) on page 735.

[31] The local clock can be stopped during USB suspend

The following table summarizes the behavior of these registers.

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

Table 182: RAM section configuration

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

See chapter Memory on page 18 for more information on RAM sections.

7.40.2 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50081000 0x40081000	APPLICATION	VMC	VMC : S VMC : NS	US	NA	Volatile memory controller	
0x41081000	NETWORK	VMC	VMC	NS	NA	Volatile memory controller	4 RAM slaves implemented 4 RAM slaves implemented

Table 183: Instances

Register	Offset	Security	Description
RAM[n].POWER	0x600		RAM[n] power control register
RAM[n].POWERSET	0x604		RAM[n] power control set register
RAM[n].POWERCLR	0x608		RAM[n] power control clear register

Table 184: Register overview

7.40.2.1 RAM[n].POWER (n=0..7)

Address offset: 0x600 + (n × 0x10)

RAM[n] power control register

[32] RAM section power off gives negligible reduction in current consumption when retention is on.

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

7.40.2.2 RAM[n].POWERSET (n=0..7)

Address offset: 0x604 + (n × 0x10)

RAM[n] power control set register

When read, this register will return the value of the RAM[n].POWER register.

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

7.40.2.3 RAM[n].POWERCLR (n=0..7)

Address offset: 0x608 + (n × 0x10)

RAM[n] power control clear register

When read, this register will return the value of the RAM[n].POWER register.

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

7.41 WDT — Watchdog timer

The countdown watchdog timer (WDT) uses the low-frequency clock source (LFCLK) and offers configurable and robust protection against application lock-up.

WDT must be configured before it is started. After configuration, WDT is started by triggering the START task.

When WDT is running, its configuration registers (CRV, RREN, and CONFIG) are blocked for further configuration.

WDT can be paused while the CPU is sleeping, or when the debugger has halted the CPU. WDT is implemented as a down-counter that generates a TIMEOUT event when it wraps over after counting down to 0. When WDT is started by 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 timeout period for the watchdog is given by the following equation:

```
timeout [s] = ( CRV + 1 ) / 32768
```

When started, WDT will make the 32.768 kHz RC oscillator start if no other 32.768 kHz clock source is running and generating the 32.768 kHz system clock, see chapter CLOCK — Clock control on page 69.

7.41.1 Reload criteria

WDT has eight separate reload request registers. These registers are used to request WDT to reload its counter with the value specified in the CRV register. To reload the watchdog counter, write `0x6E524635` to all enabled reload registers.

One or more RR registers can be individually enabled through the RREN register.

7.41.2 Temporarily pausing the watchdog

By default, the watchdog will be active counting down the down-counter while the CPU is sleeping. It is possible to configure the watchdog to automatically pause when the CPU is sleeping or when it is stopped by the debugger.

Entering System OFF mode will stop and disable the watchdog.

7.41.3 Watchdog reset

A TIMEOUT event automatically leads to a watchdog reset.

If the watchdog is configured to generate an interrupt on the TIMEOUT event, the watchdog reset is postponed by two 32.768 kHz clock cycles after the TIMEOUT event is generated. Once the TIMEOUT event is generated, and unless the watchdog is stopped, the impending watchdog reset will occur.

The watchdog can be reset from several reset sources, see Application core reset behavior on page 64. After a reset, the watchdog configuration registers are available for configuration.

See RESET — Reset control on page 61 for more information about reset sources.

7.41.4 Stopping the watchdog

By default, the watchdog cannot be stopped. It is possible to configure the watchdog to allow the STOP task.

To stop the watchdog, perform the following steps.

1. Set the CONFIG register's STOPEN field to `Enable` during watchdog configuration.
2. Write the special value 0x6E524635 to the TSEN register.
3. Invoke the STOP task.

When these conditions are met, the watchdog is stopped and a STOPPED event is issued.

When the watchdog is stopped, its configuration registers CRV, RREN, and CONFIG are no longer blocked.

> **Note:** It is recommended to write zeros to TSEN on page 743 after the watchdog has stopped, to avoid runaway code triggering the STOP task.

7.41.5 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50018000 0x40018000	APPLICATION	WDT	WDT0 : S WDT0 : NS	US	NA	Watchdog timer 0	
0x50019000 0x40019000	APPLICATION	WDT	WDT1 : S WDT1 : NS	US	NA	Watchdog timer 1	
0x4100B000	NETWORK	WDT	WDT	NS	NA	Watchdog timer	

Table 185: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start WDT
TASKS_STOP	0x004		Stop WDT
SUBSCRIBE_START	0x080		Subscribe configuration for task START
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP
EVENTS_TIMEOUT	0x100		Watchdog timeout
EVENTS_STOPPED	0x104		Watchdog stopped
PUBLISH_TIMEOUT	0x180		Publish configuration for event TIMEOUT
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
NMIENSET	0x324		Enable interrupt
NMIENCLR	0x328		Disable interrupt
RUNSTATUS	0x400		Run status
REQSTATUS	0x404		Request status
CRV	0x504		Counter reload value
RREN	0x508		Enable register for reload request registers
CONFIG	0x50C		Configuration register
TSEN	0x520		Task stop enable
RR[n]	0x600		Reload request n

Table 186: Register overview

7.41.5.1 TASKS_START

Address offset: 0x000

Start WDT

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start WDT
			Trigger	1	Trigger task

7.41.5.2 TASKS_STOP

Address offset: 0x004

Stop WDT

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop WDT
			Trigger	1	Trigger task

7.41.5.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task 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				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.41.5.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

7.41.5.5 EVENTS_TIMEOUT

Address offset: 0x100

Watchdog timeout

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_TIMEOUT			Watchdog timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.41.5.6 EVENTS_STOPPED

Address offset: 0x104

Watchdog stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			Watchdog stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.41.5.7 PUBLISH_TIMEOUT

Address offset: 0x180

Publish configuration for event TIMEOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event TIMEOUT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.41.5.8 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B 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	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[255..0]	DPPI channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

7.41.5.9 INTENSET

Address offset: 0x304

Enable interrupt

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

7.41.5.10 INTENCLR

Address offset: 0x308

Disable interrupt

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

7.41.5.11 NMIENSET

Address offset: 0x324

Enable interrupt

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

7.41.5.12 NMIENCLR

Address offset: 0x328

Disable interrupt

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

ID	R/W	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to disable interrupt for event TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.41.5.13 RUNSTATUS

Address offset: 0x400

Run status

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

ID	R/W	Field	Value ID	Value	Description
A	R	RUNSTATUSWDT			Indicates whether or not WDT is running
			NotRunning	0	Watchdog is not running
			Running	1	Watchdog is running

7.41.5.14 REQSTATUS

Address offset: 0x404

Request status

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

ID	R/W	Field	Value ID	Value	Description
A-H	R	RR[i] (i=0..7)			Request status for RR[i] register
			DisabledOrRequested	0	RR[i] register is not enabled, or are already requesting reload
			EnabledAndUnrequested	1	RR[i] register is enabled, and are not yet requesting reload

7.41.5.15 CRV

Address offset: 0x504

Counter reload value

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	CRV		[0xF..0xFFFFFFFF]	Counter reload value in number of cycles of the 32.768 kHz clock

7.41.5.16 RREN

Address offset: 0x508

Enable register for reload request registers

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E D C B A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A-H	RW	RR[i] (i=0..7)			Enable or disable RR[i] register
			Disabled	0	Disable RR[i] register
			Enabled	1	Enable RR[i] register

7.41.5.17 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				F C A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	SLEEP			Configure WDT to either be paused, or kept running, while the CPU is sleeping
			Pause	0	Pause WDT while the CPU is sleeping
			Run	1	Keep WDT running while the CPU is sleeping
C	RW	HALT			Configure WDT to either be paused, or kept running, while the CPU is halted by the debugger
			Pause	0	Pause WDT while the CPU is halted by the debugger
			Run	1	Keep WDT running while the CPU is halted by the debugger
F	RW	STOPEN			Allow stopping WDT
			Disable	0	Do not allow stopping WDT
			Enable	1	Allow stopping WDT

7.41.5.18 TSEN

Address offset: 0x520

Task stop 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 A A A A A A 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	R/W	Field	Value ID	Value	Description
A	W	TSEN			Allow stopping WDT
			Enable	0x6E524635	Value to allow stopping WDT

7.41.5.19 RR[n] (n=0..7)

Address offset: 0x600 + (n × 0x4)

Reload request n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	RR			Reload request register
			Reload	0x6E524635	Value to request a reload of the watchdog timer

7.41.6 Electrical specification

7.41.6.1 Watchdog Timer Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{WDT}	Time out interval	..	..	..	

8 Debug and trace

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

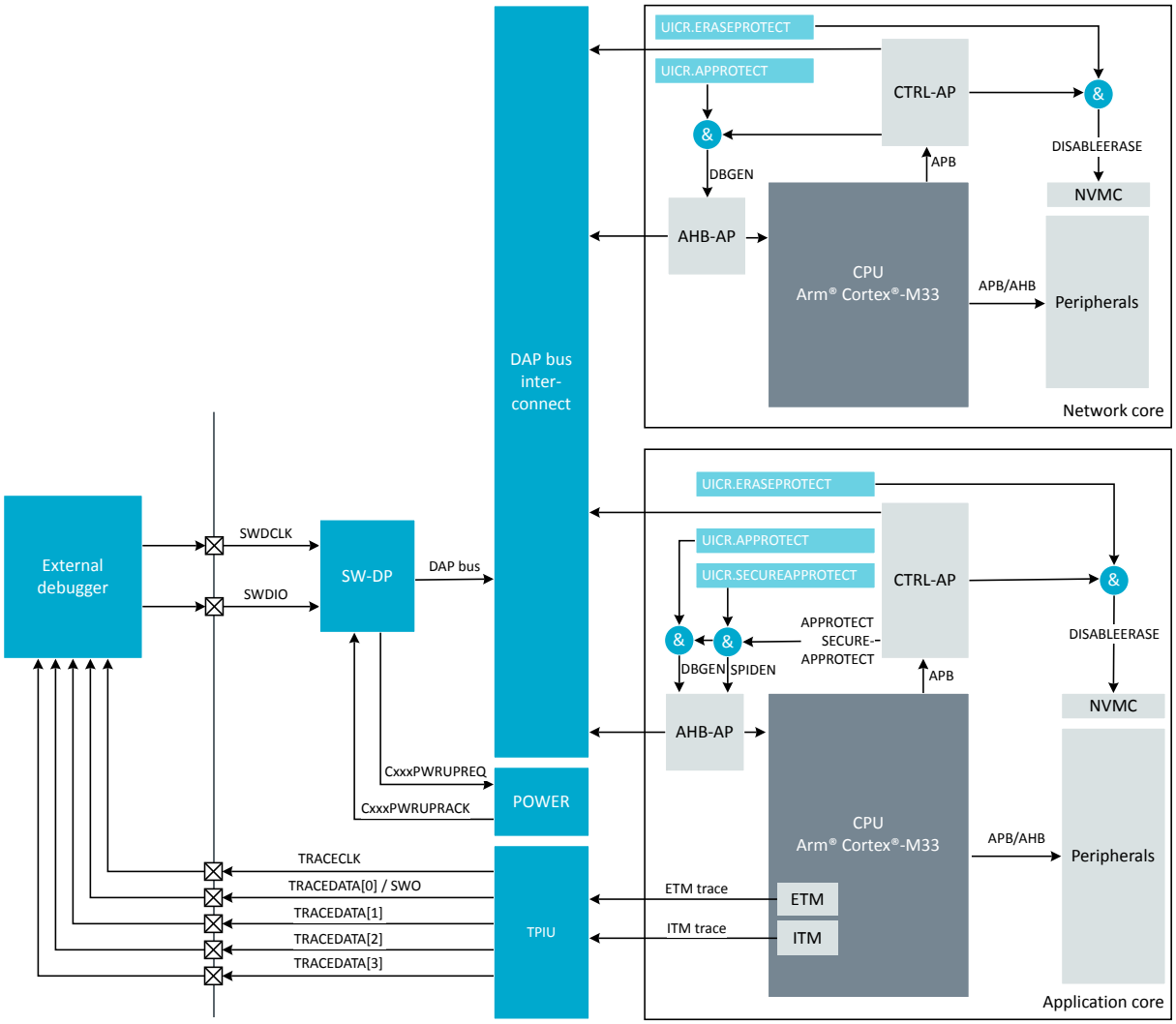

Figure 245: Debug and trace overview

The main features of the debug and trace system are:

- Access port connection to application core Arm Cortex-M33
  - Eight breakpoints
  - Four watchpoint comparators
  - Instrumentation trace macrocell (ITM)
  - Embedded trace macrocell (ETM)
  - Access protection through APPROTECT, ERASEPROTECT and SECUREAPPROTECT
- Access port connection to network core Arm Cortex-M33
  - Eight breakpoints
  - Four watchpoints
  - Access protection through APPROTECT and ERASEPROTECT
- Serial wire debug (SWD) interface protocol version 2 with multidrop support
- Trace port interface unit (TPIU)

- 4-bit parallel trace of ITM and ETM trace data
- Serial wire output (SWO) trace of ITM data

8.1 DAP — Debug access port

An external debugger can access the device using the DAP.

The DAP is a standard Arm CoreSight™ serial wire debug port (SW-DP) that implements the serial wire debug (SWD) protocol – a two-pin serial interface using SWDCLK and SWDIO pins (see Debug and trace overview on page 745).

Note:

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

There are several access ports that connect to different parts of the system. See the following table for more information.

AP ID	Type	Description
0	AHB-AP	Application subsystem access port
1	AHB-AP	Network subsystem access port
2	CTRL-AP	Application subsystem control access port
3	CTRL-AP	Network subsystem control access port

Table 187: Access port overview

The AHB-AP and APB-AP access ports are standard Arm components documented in the *Arm CoreSight SoC-400 Technical Reference Manual, Revision r3p2*. The CTRL-AP access port is proprietary (see CTRL-AP - Control access port on page 757).

8.1.1 Registers

Register	Offset	Security	Description
TARGETID	0x042		The TARGETID register provides information about the target when the host is connected to a single device. The TARGETID register is accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x2.
DLPIDR	0x043		The DLPIDR register provides information about the serial wire debug protocol version. Accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x3.

Table 188: Register overview

8.1.1.1 TARGETID

Address offset: 0x042

The TARGETID register provides information about the target when the host is connected to a single device.

The TARGETID register is accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x2.

Bit number				31	30	29	28	27	26	25	24	23	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	B	B	B	B	B	B	B	B	B	B	B	B					A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x30070289				**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
ID	R/W	Field	Value ID	Value																															
A	R	TDESIGNER																																	
			NordicSemi	0x144																															
B	R	TPARTNO																																	
			nRF53	7																															
C	R	TREVISION																																	
			nRF5340LS	2																															
			nRF5340	3																															

8.1.1.2 DLPIDR

Address offset: 0x043

The DLPIDR register provides information about the serial wire debug protocol version.

Accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x3.

Bit number				31	30	29	28	27	26	25	24	23	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	A	A	A
Reset 0x00000001				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
ID	R/W	Field	Value ID	Value																															
A	R	PROTVSN																																	
			SWDPv2	1																															
B	R	TINSTANCE																																	

This value is set by the UICR.TINSTANCE register.

8.1.2 Electrical specification

8.1.2.1 SW-DP

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

8.1.2.2 Trace port

Symbol	Description	Min.	Typ.	Max.	Units
T_{cyc}	Clock period, as defined by Arm in Embedded Trace Macrocell Architecture Specification (see Timing specifications in Trace Port Physical Interface section)	15.625		250	ns

8.2 Access port protection

The control access ports are always accessible from the debugger, while access to the system resources through each core's individual access ports (AHB-AP) can be protected in different ways.

The following tables give an overview of the access port protection methods.

Registers	Description
UICR.APPROTECT and CTRL-AP.APPROTECT.DISABLE	These registers control the generation of the application core AHB-AP DBGEN signal, which controls all non-secure access through the application core AHB-AP. This can be used to provide readback protection of the flash contents. See also Application core access port protection for non-secure debug access on page 749. For more information about the DBGEN signal, see the *Arm CoreSight SoC-400 Technical Reference Manual, Revision r3p2*.
UICR.SECUREAPPROTECT and CTRL-AP.SECUREAPPROTECT.DISABLE	These registers control the generation of the application core AHB-AP SPIDEN signal, which blocks all secure access through the application core AHB-AP. This means that only the non-secure code can be debugged and accessed. To enable access to the secure access port, APPROTECT must be unprotected. See also Application core access port protection for secure debug access on page 749. For more information about the SPIDEN signal, see the *Arm CoreSight SoC-400 Technical Reference Manual, Revision r3p2*.
UICR.ERASEPROTECT and CTRL-AP.ERASEPROTECT.DISABLE	Disables the application core CTRL-AP.ERASEALL and NVMC ERASEALL functionality. This can be used together with APPROTECT to provide read-back and re-purposing protection.

Table 189: Application core access port protection overview

Registers	Description
UICR.APPROTECT and CTRL-AP.APPROTECT.DISABLE	These registers control the generation of the network core AHB-AP DBGEN signal, which blocks all access through the network core AHB-AP. See also Network core access port protection for debug access on page 749. For the network core that does not feature TrustZone, only DBGEN can be controlled and SPIDEN is not used.
UICR.ERASEPROTECT	Disables the network core CTRL-AP.ERASEALL and NVMC ERASEALL functionality. This can be used together with APPROTECT to provide read-back and re-purposing protection.

Table 190: Network core access port protection overview

For both cores, UICR and CTRL-AP are combined to enable or disable the access port protection. The access port is normally protected, and is opened when the following conditions are met:

1. UICR.APPROTECT must be `Unprotected`.

2. CTRL-AP.APPROTECT.DISABLE on both CPU and debugger side must match. However, after reset the debugger side register value is known and CPU can open the port by writing `Unprotected` to the register.

The following tables lists the available APPROTECT combinations.

Application core UICR.APPROTECT	CPU and debugger side CTRL-AP.APPROTECT.DISABLE registers are equal	DBGEN	Debug access to application core AHB-AP
`Protected`	No	0	Not permitted
`Protected`	Yes	0	Not permitted
`Unprotected`	No	0	Not permitted
`Unprotected`	Yes	1	Permitted

Table 191: Application core access port protection for non-secure debug access

Application core UICR.SECUREAPPROTECT	CPU and debugger side CTRL-AP.SECUREAP-PROTECT.DISABLE registers are equal	SPIDEN	Secure debug access to application core AHB-AP
`Protected`	No	0	Not permitted
`Protected`	Yes	0	Not permitted
`Unprotected`	No	0	Not permitted
`Unprotected`	Yes	1	Permitted

Table 192: Application core access port protection for secure debug access

Network core UICR.AP-PROTECT	CPU and debugger side CTRL-AP.APPROTECT registers are equal	DBGEN	Debug access to AHB-AP
`Protected`	No	0	Not permitted
`Protected`	Yes	0	Not permitted
`Unprotected`	No	0	Not permitted
`Unprotected`	Yes	1	Permitted

Table 193: Network core access port protection for debug access

The access port is also open after the completion of the CTRL-AP.ERASEALL operation. After completing the erase operation, CTRL-AP will temporarily unprotect AHB-AP. AHB-AP will be protected when one of the following conditions are met:

- Power-on reset
- Brown-out reset
- Watchdog timer reset
- Pin reset

The following figure is an example on how nRF5340 with access port protection enabled can be erased, programmed, and configured to allow debugging. Operations sent from debugger as well as registers written by firmware will affect the access port state. The operation named Reset* is one of the conditions listed above.

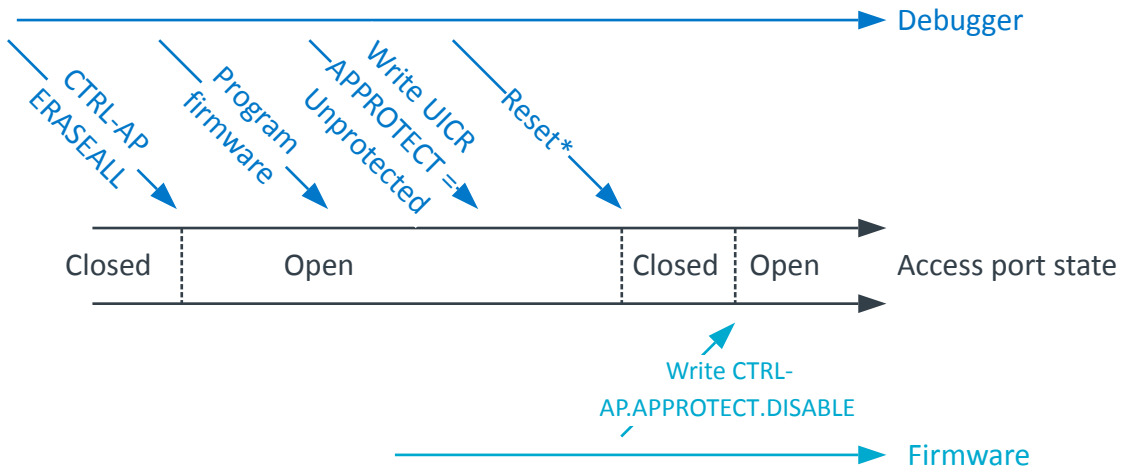

Figure 246: Access port unlocking

The debugger can read the access port protection status in the core's AHB-AP, using the Arm AHB-AP Control/Status Word register (CSW), defined in the *Arm CoreSight SoC-400 Technical Reference Manual, Revision r3p2*. The DbgStatus field indicates that the AHB-AP can perform AHB transfers, while the SPIStatus field indicates if secure AHB transfers are permitted. For a list of all debug access ports, see DAP — Debug access port on page 746.

For more details on CTRLAP.ERASEALL, CTRLAP.SECUREAPPROTECT, and CTRLAP.APPROTECT, see CTRL-AP - Control access port on page 757.

Note: Using SPU — System protection unit on page 584, the application core can be configured to grant the network core access to its resources. This grant also applies to the network core AHB-AP.

8.3 Debug Interface mode

Before the external debugger can use any of the access ports, 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. Otherwise, the device is in Normal mode. When a debug session is over, the external debugger must return the device back to Normal mode and perform a pin reset. This is due to the overall power consumption being higher in Debug interface mode compared to Normal mode.

Some peripherals behave differently in Debug interface mode compared to Normal mode. These differences are described in more detail in the chapters of the affected peripherals.

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

If the device is in System OFF when power is requested via CxxxPWRUPREQ, the device will wake up and the DIF flag in RESETREAS on page 66 will be set.

8.4 Real-time debug

The device supports real-time debugging, which allows interrupts to execute to completion in real time when breakpoints are set in Thread mode or lower priority interrupts.

Real-time debugging enables breakpoint setting and single-stepping through code without causing the failure of 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 stepping through code in a low-priority thread.

8.5 ROM tables

Each ROM Table on the SoC contains a listing of the components that are connected to the debug port or AHB-AP. These listings allow an external debugger or on-chip software to discover the CoreSight devices on the SoC.

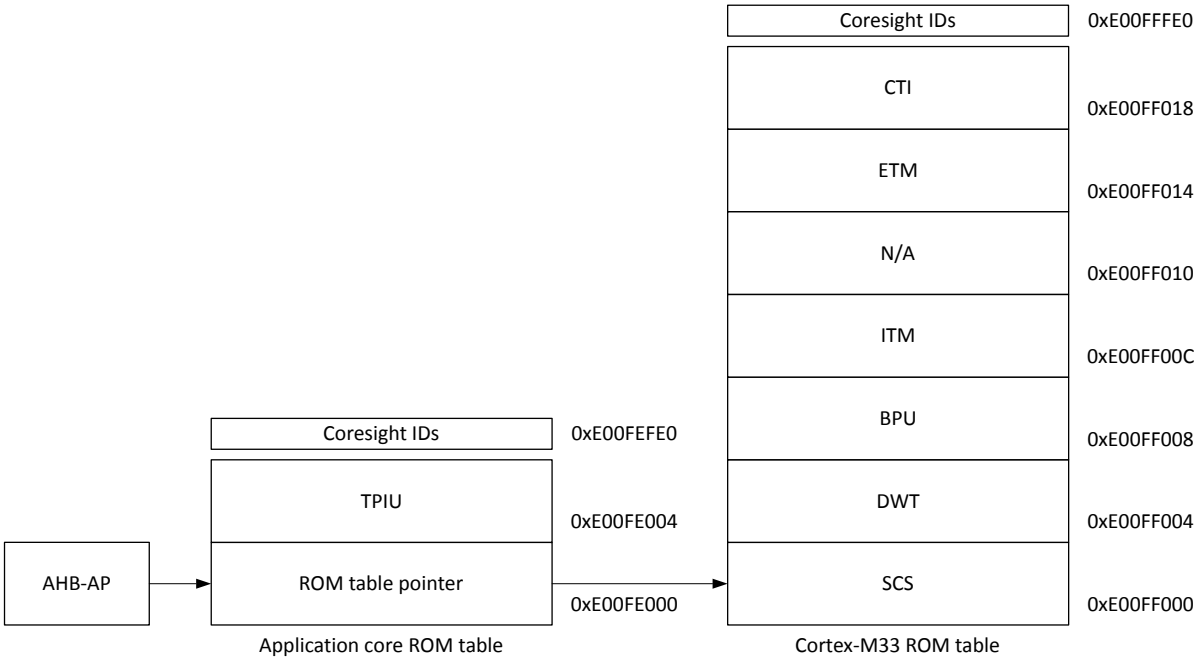

Figure 247: Application core ROM table overview

Address	Component	Value
0xE00FEFFC	CIDR3	0x000000B1
0xE00FEFF8	CIDR2	0x00000005
0xE00FEFF4	CIDR1	0x00000010
0xE00FEFF0	CIDR0	0x0000000D
0xE00FEFDC	PIDR7	0x00000000
0xE00FEFD8	PIDR6	0x00000000
0xE00FEFD4	PIDR5	0x00000000
0xE00FEFD0	PIDR4	0x00000002
0xE00FEFEC	PIDR3	0x00000000
0xE00FEFE8	PIDR2	The following revisions are defined: • 0x0000002C nRF5340 EngA • 0x0000003C nRF5340 EngB • 0x0000004C nRF5340 EngC • 0x0000005C nRF5340 EngD
0xE00FEFE4	PIDR1	0x00000040
0xE00FEFE0	PIDR0	0x00000007
0xE00FEFCC	MEMTYPE	0x00000001
0xE00FE004	TPIU	0xFFF42003
0xE00FE000	ROM table	0x00001003

Table 194: Application core ROM table entries

Address	Component	Value
0xE00FF01C	MTB (not implemented)	0xFFF44002
0xE00FF018	CTI	0xFFF43003
0xE00FF014	ETM	0xFFF42003
0xE00FF00C	ITM	0xFFF01003
0xE00FF008	BPU	0xFFF03003
0xE00FF004	DWT	0xFFF02003
0xE00FF000	SCS	0xFFF0F003

Table 195: Application Arm Cortex-M33 ROM table entries

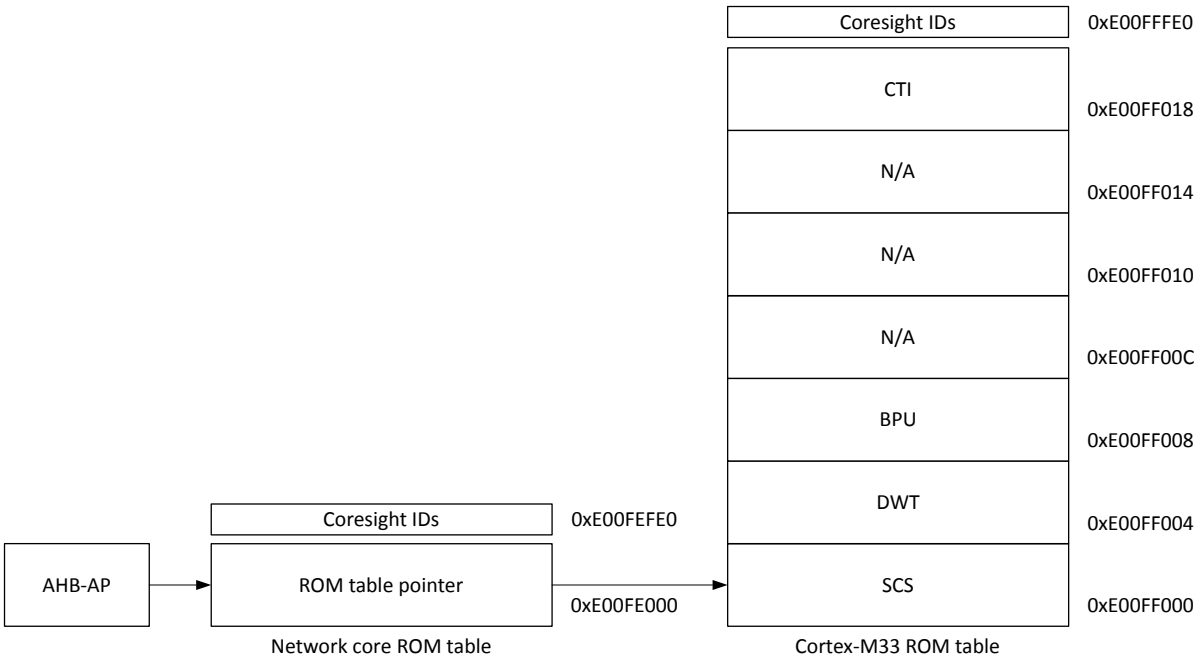

Figure 248: Network core ROM table overview

Address	Component	Value
0xE00FEFFC	CIDR3	0x000000B1
0xE00FEFF8	CIDR2	0x00000005
0xE00FEFF4	CIDR1	0x00000010
0xE00FEFF0	CIDR0	0x0000000D
0xE00FEFDC	PIDR7	0x00000000
0xE00FEFD8	PIDR6	0x00000000
0xE00FEFD4	PIDR5	0x00000000
0xE00FEFD0	PIDR4	0x00000002
0xE00FEFEC	PIDR3	0x00000000
0xE00FEFE8	PIDR2	The following revisions are defined: • 0x0000002C nRF5340 EngA • 0x0000003C nRF5340 EngB • 0x0000004C nRF5340 EngC • 0x0000005C nRF5340 EngD
0xE00FEFE4	PIDR1	0x00000040
0xE00FEFE0	PIDR0	0x00000007
0xE00FEFCC	MEMTYPE	0x00000001
0xE00FE000	ROM table	0x00001003

Table 196: Network core ROM table entries

Address	Component	Value
0xE00FF01C	MTB (not implemented)	0xFFF44002
0xE00FF018	CTI	0xFFF43003
0xE00FF014	ETM (not implemented)	0xFFF42002
0xE00FF00C	ITM (not implemented)	0xFFF01002
0xE00FF008	BPU	0xFFF03003
0xE00FF004	DWT	0xFFF02003
0xE00FF000	SCS	0xFFF0F003

Table 197: Network Arm Cortex-M33 ROM table entries

8.6 Cross-trigger network

The debug system has a cross-trigger network used to simultaneously start and halt the cores in the system.

The following diagram shows an overview of the cross-trigger connections.

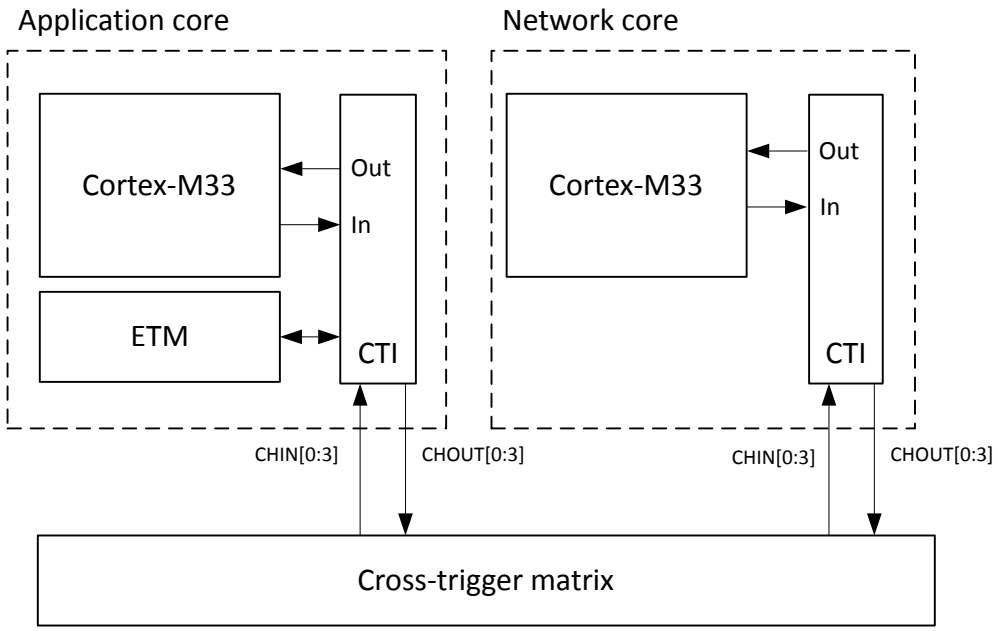

Figure 249: Cross-trigger network block diagram

Both the application and network cores have a cross-trigger interface (CTI) peripheral that can trigger events or be triggered by signals in the processor or debug blocks. The CTI can be configured to route trigger in-signals to trigger out-signals within the CTI or the cross-trigger matrix. The cross-trigger matrix has four channels in total that can be used to communicate trigger signals between cores.

You can stop the network core when the application core is stopped (due to a breakpoint or a stopped debug session), by doing the following:

1. Configure the application core CTI to generate an event on channel 0 for CTITRIGIN[0] (processor halted) using CTIINEN[0].
2. Configure the network core CTI to trigger CTITRIGOUT[0] (processor debug request) on channel 0 using CTIOUTEN[0].

Configuring the cross-trigger interface

In this example, the following CTI channels are used:

- Channel 0 is used to relay debug requests from the application to the network domain.

- Channel 1 is used to relay debug requests from the network to the application domain.
- Channel 2 is used by the debugger to send a common trigger for restarting both domains after a breakpoint.

For the application core, add the following code:

```
#define CTI_TRIGIN_CPUHALTED   0
#define CTI_TRIGOUT_DEBUGREQ   0
#define CTI_TRIGOUT_CPURESTART 1
...
// Enable global CTI routing
NRF_CTI_S->CTICONTROL = CTI_CTICONTROL_GLBEN_Enabled;
// Connect the CPU halted trigger of this domain to debug request of the other domain
NRF_CTI_S->CTIINEN[CTI_TRIGIN_CPUHALTED]    = CTI_CTIINEN_TRIGINEN_0_Msk;
NRF_CTI_S->CTIOUTEN[CTI_TRIGOUT_DEBUGREQ]   = CTI_CTIOUTEN_TRIGOUTEN_1_Msk;
NRF_CTI_S->CTIOUTEN[CTI_TRIGOUT_CPURESTART] = CTI_CTIOUTEN_TRIGOUTEN_2_Msk;
```

For the network core, add the following code:

```
#define CTI_TRIGIN_CPUHALTED   0
#define CTI_TRIGOUT_DEBUGREQ   0
#define CTI_TRIGOUT_CPURESTART 1
...
// Enable global CTI routing
NRF_CTI_NS->CTICONTROL = CTI_CTICONTROL_GLBEN_Enabled;
// Connect the CPU halted trigger of this domain to debug request of the other domain
NRF_CTI_NS->CTIINEN[CTI_TRIGIN_CPUHALTED]    = CTI_CTIINEN_TRIGINEN_1_Msk;
NRF_CTI_NS->CTIOUTEN[CTI_TRIGOUT_DEBUGREQ]   = CTI_CTIOUTEN_TRIGOUTEN_0_Msk;
NRF_CTI_NS->CTIOUTEN[CTI_TRIGOUT_CPURESTART] = CTI_CTIOUTEN_TRIGOUTEN_2_Msk;
```

See the following tables for more information about trigger connections to and from the CTI.

Signal	Description
CTITRIGIN[0]	Processor halted
CTITRIGIN[1]	DWT comparator output 0
CTITRIGIN[2]	DWT comparator output 1
CTITRIGIN[3]	DWT comparator output 2
CTITRIGIN[4]	ETM event output 0
CTITRIGIN[5]	ETM event output 1

Table 198: Application core triggers to CTI

Signal	Description
CTITRIGOUT[0]	Processor debug request
CTITRIGOUT[1]	Processor restart
CTITRIGOUT[2]	N/A
CTITRIGOUT[3]	N/A
CTITRIGOUT[4]	ETM event input 0
CTITRIGOUT[5]	ETM event input 1
CTITRIGOUT[6]	ETM event input 2
CTITRIGOUT[7]	ETM event input 3

Table 199: Application core triggers from CTI

Signal	Description
CTITRIGIN[0]	Processor halted
CTITRIGIN[1]	DWT comparator output 0
CTITRIGIN[2]	DWT comparator output 1
CTITRIGIN[3]	DWT comparator output 2

Table 200: Network core triggers to CTI

Signal	Description
CTITRIGOUT[0]	Processor debug request
CTITRIGOUT[1]	Processor restart

Table 201: Network core triggers from CTI

8.7 Multidrop serial wire debug

Multidrop serial wire debug allows simultaneous access to an unlimited number of devices through a single connection. This is useful for connectivity-constrained products that contain multiple chips with multidrop support.

In order to select a target in a multidrop capable product, the debugger must write the correct TINSTANCE, TPARTNO, and TDESIGNER fields into the SW-DP TARGETSEL register. The values for these fields are located in and fetched from two registers, TARGETID on page 746 and DLPIDR on page 747.

For more information about multidrop SWD, see *Arm Debug Interface Architecture Specification*, ADIv5.0 to ADIv5.2.

8.8 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 Debug and trace overview on page 745 (TRACEDATA[0] through TRACEDATA[3], and TRACECLK).

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 supported in Parallel trace mode only, while both parallel and Serial trace modes support the ITM trace. For details on how to use the trace capabilities, see the debug documentation of your IDE.

TPIU's dedicated trace pins are multiplexed with GPIOs. SWO and TRACEDATA[0] use the same GPIO. Trace is limited to dedicated pins. See Pin assignments on page 783 for more information.

Trace speed is configured in the register TRACEPORTSPEED (Retained) on page 782. The speed of the trace pins depends on the drive setting of the GPIOs that the trace pins are multiplexed with. The drive setting is configured using the DRIVE field of the GPIO register PIN_CNF[n] (n=0..31) (Retained) on page 228.

Only drive settings S0S1, H0H1, and E0E1 should be used 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), or extra high drive (E0E1) for the fastest trace port speeds. Ensure that the drive setting of the GPIOs are not overwritten by software during the debugging session.

In addition to DRIVE, the GPIO pin must be assigned to trace and debug (TND), using the MCUSEL field of the the PIN_CNF register. When pins are assigned to TND, these GPIOs are output-only, and other functionality of the pins is disabled.

8.9 Enabling the trace port

A specific sequence of operations must be performed to enable the trace port.

1. Enable the debug master.

```
NRF_TAD_S->ENABLE = TAD_ENABLE_ENABLE_Msk;
```

2. Request clock startup.

```
NRF_TAD_S->CLOCKSTART = TAD_CLOCKSTART_START_Msk;
```

3. Configure the trace port to use pins P0.08 through P0.12.

```
NRF_TAD_S->PSEL.TRACECLK   = TAD_PSEL_TRACECLK_PIN_Traceclk;
NRF_TAD_S->PSEL.TRACEDATA0 = TAD_PSEL_TRACEDATA0_PIN_Tracedata0;
NRF_TAD_S->PSEL.TRACEDATA1 = TAD_PSEL_TRACEDATA1_PIN_Tracedata1;
NRF_TAD_S->PSEL.TRACEDATA2 = TAD_PSEL_TRACEDATA2_PIN_Tracedata2;
NRF_TAD_S->PSEL.TRACEDATA3 = TAD_PSEL_TRACEDATA3_PIN_Tracedata3;
```

4. Configure the GPIO pins so that the trace and debug system can control them. Set high drive strength to ensure sufficiently fast operation. Do this for all trace pins that should be used.

```
// Clear the bitfields before configuring to make sure the correct value is written
NRF_P0_S->PIN_CNF[TAD_PSEL_TRACECLK_PIN_Traceclk]
                    &= ~(GPIO_PIN_CNF_MCUSEL_Msk | GPIO_PIN_CNF_DRIVE_Msk);
NRF_P0_S->PIN_CNF[TAD_PSEL_TRACECLK_PIN_Traceclk]
                    |= (GPIO_PIN_CNF_MCUSEL_TND << GPIO_PIN_CNF_MCUSEL_Pos)
                     | (GPIO_PIN_CNF_DRIVE_E0E1 << GPIO_PIN_CNF_DRIVE_Pos);
```

5. Set trace port speed to 64 MHz.

```
NRF_TAD_S->TRACEPORTSPEED = TAD_TRACEPORTSPEED_TRACEPORTSPEED_64MHz;
```

Note: Although possible, it is not recommended to run the trace port at less than half the CPU frequency, as it risks dropping some trace packets.

6. Configure Arm CoreSight components (see Arm CoreSight documentation for more information).

8.10 CTRL-AP - Control access port

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

For an overview of the other debug access ports, see DAP — Debug access port on page 746.

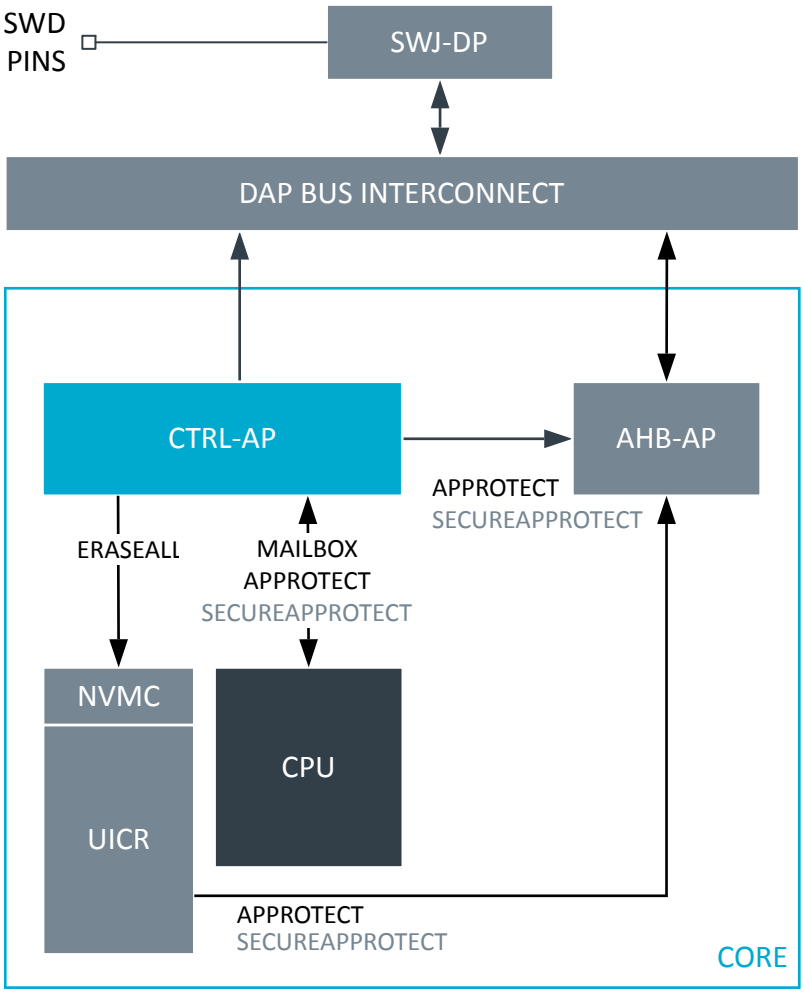

Figure 250: Control access port details

Access port protection (APPROTECT) blocks the debugger access to the AHB-AP, and prevents read and write access to all CPU registers and memory-mapped addresses. To enable port protection access for both secure and non-secure modes, use the registers UICR.SECUREAPPROTECT and UICR.APPROTECT , as well as CTRLAP.APPROTECT.DISABLE and CTRLAP.SECUREAPPROTECT.DISABLE. The debugger can use the register to read the status of secure and non-secure access port protection.

Erase protection (ERASEPROTECT) protects the flash and UICR parts of the non-volatile memory from being erased. Erase protection can be temporarily disabled from the control access port.

CTRL-AP has the following features:

- Soft reset
- Erase all
- Mailbox interface
- Debug of protected devices

8.10.1 Reset request

The debugger can request the device to perform a soft reset.

Use the register RESET on page 761 to request a soft reset. Once the soft reset is performed, the reset reason is accessible on the on-chip firmware through the RESETREAS register. For more information about the soft reset, see RESET — Reset control on page 61.

8.10.2 Erase all

The erase all function lets the debugger trigger an erase of flash, user information configuration registers (UICR), RAM, all peripheral settings, and also temporarily removes the access port protection.

To trigger an erase all function, the debugger writes to the register ERASEALL on page 762. The register ERASEALLSTATUS on page 762 will read as busy for the duration of the operation. The ERASEALL mechanism completes its tasks by writing UICR.APPROTECT to the `Unprotected` value, in addition to writing the CPU side CTRLAP.SECUREAPPROTECT.DISABLE and CTRLAP.APPROTECT.DISABLE registers to the value `0x50FA50FA`. After the next soft reset, the access port protection is temporarily removed. This temporary unprotection is removed by the next pin reset, power-on reset, brown-out reset, or watchdog timer reset. For more information about access port protection, see Access port protection on page 748.

If the debugger performs an erase all function on a slave MCU, the erase sequence will always erase the application MCU first, independently of how the application is protected, before erasing the slave MCU.

Erase all protection

It is possible to prevent the debugger from performing an erase all operation by writing to the UICR.ERASEPROTECT register. Once the register is configured and the device is reset, the CTRL-AP ERASEALL operation is disabled, and all flash write and erase operations are restricted to the firmware. In addition, it is still possible to write/erase from the debugger as long as the UICR.APPROTECT register is not set.

Note: Setting the UICR.ERASEPROTECT register only affects the erase all operation and not the debugger access.

The register ERASEPROTECT.STATUS on page 763 holds the status for erase protection.

8.10.3 Mailbox interface

CTRL-AP implements a mailbox interface which enables the CPU to communicate with a debugger over the SWD interface.

The mailbox interface consists of a transmit register MAILBOX.TXDATA on page 764 with its corresponding status register MAILBOX.TXSTATUS on page 764, and a receive register MAILBOX.RXDATA on page 764 with its corresponding status register MAILBOX.RXSTATUS on page 764. Status bits in the TXSTATUS/RXSTATUS registers are set and cleared automatically when the TXDATA/RXDATA registers are written to and read from, independently of the direction.

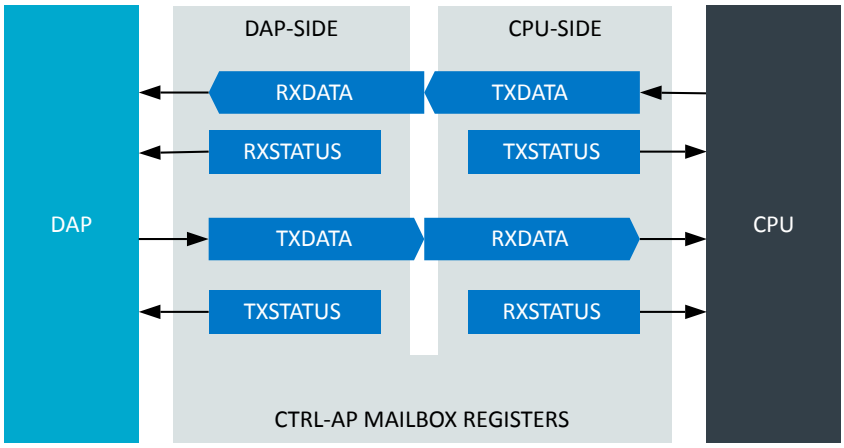

Figure 251: Mailbox register interface

Mailbox transfer sequence

1. Sender writes TXDATA.
2. Hardware sets sender's TXSTATUS to DataPending.
3. Hardware sets receiver's RXSTATUS to DataPending.
4. Receiver reads RXDATA.
5. Hardware sets receiver's RXSTATUS to NoDataPending.
6. Hardware sets sender's TXSTATUS to NoDataPending.

8.10.4 Disabling erase protection

The erase protection mechanism can be disabled to return a device to factory default settings on next reset.

The debugger can read the erase protection status in the register ERASEPROTECT.STATUS on page 763.

If ERASEPROTECT has been enabled, both the debugger and on-chip firmware must write the same non-zero 32-bit KEY value into their respective ERASEPROTECT.DISABLE registers to disable the erase protection. When both registers have been written with the same non-zero 32-bit KEY value, the device is automatically erased as described in Erase all on page 759. The access ports will be re-enabled on the next reset once the secure erase sequence has completed.

The write-once register ERASEPROTECT.LOCK on page 767 should be set to `Locked` as early as possible in the start-up sequence, preferably as soon as the on-chip firmware has determined it does not need to communicate with a debugger over the CTRL-AP mailbox interface. Once written, it will not be possible to remove the erase protection until the next reset.

8.10.5 Disabling access port protection

The access port protection mechanisms can be temporarily disabled to debug the device.

The disabling of the access port protection is done through a combination of UICR and CTRL-AP registers.

Disabling non-secure access port protection

If UICR.APPROTECT has been enabled from UICR, the device access port is protected.

If UICR.APPROTECT has not been enabled from UICR, both the debugger and on-chip firmware must write the same non-zero 32-bit KEY value into their respective registers CTRLAP.APPROTECT.DISABLE (CPU-side) and CTRLAP.APPROTECT.DISABLE (debugger-side) to disable the access port protection to non-secure mode.

The write-once register APPROTECT.LOCK on page 767 should be set to `Locked` as early as possible in the start-up sequence. Once written, it will not be possible to remove the non-secure mode access port protection until next reset.

Disabling secure access port protection

If UICR.APPROTECT has been enabled from UICR, the device access port is protected.

If UICR.SECUREAPPROTECT has not been enabled from UICR, both the debugger and on-chip firmware must write the same non-zero 32-bit KEY value into their respective registers CTRLAP.SECUREAPPROTECT.DISABLE (CPU-side) and CTRLAP.SECUREAPPROTECT.DISABLE (debugger-side) to disable the access port protection to secure mode.

The write-once register SECUREAPPROTECT.LOCK on page 768 should be set to `Locked` as early as possible in the start-up sequence, preferably as soon as on-chip firmware has determined it does not need to communicate with a debugger over the CTRL-AP mailbox interface. Once written, it will not be possible to remove the secure mode access port protection until next reset.

Note: If secure mode debug is enabled, an ERASEALL sequence can also be initiated by writing the same 32-bit KEY value into the respective ERASEPROTECT.DISABLE registers

The CTRLAP.APPROTECT.DISABLE and CTRLAP.SECUREAPPROTECT.DISABLE registers are only reset by pin reset, brown-out reset, or watchdog timer reset. This allows keeping the debug session active through soft resets.

After an ERASEALL sequence has completed, the access port protection of the core's AHB-AP is disabled until the next pin reset, power-on reset, brown-out reset, or watchdog timer reset. This will allow initial firmware to be written. For more details on ERASEALL, see Erase all on page 759.

Access port protection status

The debugger can read the access port protection status in the core's AHB-AP, using the Arm AHB-AP Control/Status Word register (CSW), defined in the *Arm CoreSight SoC-400 Technical Reference Manual, Revision r3p2*. The `DbgStatus` field indicates that the AHB-AP can perform AHB transfers, while the `SPIStatus` indicates if secure AHB transfers are permitted. For a list of all debug access ports, see DAP — Debug access port on page 746.

8.10.6 Debugger registers

CTRL-AP has a set of registers that can only be accessed from the debugger over the SWD interface. These are not accessible from the CPU.

The SECUREAPPROTECT fields and registers only apply for cores that have the Arm Cortex-M33 with TrustZone technology.

8.10.6.1 Registers

Register	Offset	Security	Description
RESET	0x000		System reset request.
ERASEALL	0x004		Perform a secure erase of the device, where flash, SRAM, and UICR will be erased in sequence. The device will be returned to factory default settings upon next reset.
ERASEALLSTATUS	0x008		This is the status register for the ERASEALL operation.
APPROTECT.DISABLE	0x010		This register disables APPROTECT and enables debug access to non-secure mode.
SECUREAPPROTECT.DISABLE	0x014		This register disables SECUREAPPROTECT and enables debug access to secure mode.
ERASEPROTECT.STATUS	0x018		This is the status register for the UICR ERASEPROTECT configuration.
ERASEPROTECT.DISABLE	0x01C		This register disables ERASEPROTECT and performs ERASEALL.
MAILBOX.TXDATA	0x020		Data sent from the debugger to the CPU.
MAILBOX.TXSTATUS	0x024		This register shows a status that indicates if data sent from the debugger to the CPU has been read.
MAILBOX.RXDATA	0x028		Data sent from the CPU to the debugger.
MAILBOX.RXSTATUS	0x02C		This register shows a status that indicates if data sent from the CPU to the debugger has been read.
IDR	0x0FC		CTRL-AP Identification Register, IDR.

Table 202: Register overview

8.10.6.1.1 RESET

Address offset: 0x000

System reset request.

This register is automatically deactivated during an ERASEALL operation.

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

ID	R/W	Field	Value ID	Value	Description
A	RW	RESET			System reset request and status
			NoReset	0	Write to release reset
					Reading '0' means reset is not active
			Reset	1	Write to hold reset
					Reading '1' means reset is active

8.10.6.1.2 ERASEALL

Address offset: 0x004

Perform a secure erase of the device, where flash, SRAM, and UICR will be erased in sequence. The device will be returned to factory default settings upon next 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	W	ERASEALL			Return device to factory default settings
			NoOperation	0	No operation
			Erase	1	Erase flash, SRAM, and UICR in sequence

8.10.6.1.3 ERASEALLSTATUS

Address offset: 0x008

This is the status register for the ERASEALL operation.

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

ID	R/W	Field	Value ID	Value	Description
A	R	ERASEALLSTATUS			Status bit for the ERASEALL operation
			Ready	0	ERASEALL is ready
			Busy	1	ERASEALL is busy (on-going)

8.10.6.1.4 APPROTECT.DISABLE

Address offset: 0x010

This register disables APPROTECT and enables debug access to non-secure mode.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x50FA50FA				**0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	KEY			Disable APPROTECT and enable debug access to non-secure mode until the next pin reset if KEY fields match. The current APPROTECT value as configured from CTRL-AP is disabled if the value of KEY fields are non-zero and the KEY fields match on both the CPU and debugger sides. To enable debug access, both CTRL-AP and UICR.APPROTECT protection needs to be disabled.

8.10.6.1.5 SECUREAPPROTECT.DISABLE

Address offset: 0x014

This register disables SECUREAPPROTECT and enables debug access to secure mode.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x50FA50FA				**0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	KEY			Disable SECUREAPPROTECT and enable debug of secure mode until the next pin reset if KEY fields match. The current SECUREAPPROTECT value as configured from CTRL-AP is disabled if the value of KEY fields are non-zero and the KEY fields match on both the CPU and debugger sides. To enable debug access, both CTRL-AP and UICR.SECUREAPPROTECT protection needs to be disabled.

8.10.6.1.6 ERASEPROTECT.STATUS

Address offset: 0x018

This is the status register for the UICR ERASEPROTECT 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	PALL			Status bit for erase protection **Note:** The reset value is auto read from the ERASEPROTECT register in UICR.
			Enabled	0	ERASEPROTECT is enabled
			Disabled	1	ERASEPROTECT is not enabled and ERASEALL can be performed

8.10.6.1.7 ERASEPROTECT.DISABLE

Address offset: 0x01C

This register disables ERASEPROTECT and performs ERASEALL.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	KEY			The ERASEALL sequence will be initiated if value of the KEY fields are non-zero and the KEY fields match on both the CPU and debugger sides.

8.10.6.1.8 MAILBOX.TXDATA

Address offset: 0x020

Data sent from the debugger to the CPU.

Writing to this register will automatically set a DataPending value in the TXSTATUS 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	Data			Data sent from debugger

8.10.6.1.9 MAILBOX.TXSTATUS

Address offset: 0x024

This register shows a status that indicates if data sent from the debugger to the CPU has been read.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	R	Status			Status of register DATA
			NoDataPending	0	No data pending in register TXDATA
			DataPending	1	Data pending in register TXDATA

8.10.6.1.10 MAILBOX.RXDATA

Address offset: 0x028

Data sent from the CPU to the debugger.

Reading from this register will automatically set a NoDataPending value in the RXSTATUS 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	R	Data			Data sent from CPU

8.10.6.1.11 MAILBOX.RXSTATUS

Address offset: 0x02C

This register shows a status that indicates if data sent from the CPU to the debugger has been read.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	R	Status			Status of register DATA
			NoDataPending	0	No data pending in register RXDATA
			DataPending	1	Data pending in register RXDATA

8.10.6.1.12 IDR

Address offset: 0x0FC

CTRL-AP Identification Register, IDR.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				E E E E D D D D C C C C C C B B B B A A A A A A A A
Reset 0x12880000				**0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	APID			AP Identification
B	R	CLASS			Access Port (AP) class
			NotDefined	0x0	No defined class
			MEMAP	0x8	Memory Access Port
C	R	JEP106ID			JEDEC JEP106 identity code
D	R	JEP106CONT			JEDEC JEP106 continuation code
E	R	REVISION			Revision

8.10.7 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50006000 0x40006000	APPLICATION	CTRLAPPERI	CTRLAP : S CTRLAP : NS	US	NSA	Control access port CPU side	
0x41006000	NETWORK	CTRLAPPERI	CTRLAP	NS	NA	Control access port CPU side	SECUREAPPROTECT.LOCK, SECUREAPPROTECT.DISABLI and STATUS.SECUREAPPROTECT registers not supported.

Table 203: Instances

Register	Offset	Security	Description
MAILBOX.RXDATA	0x400		Data sent from the debugger to the CPU.
MAILBOX.RXSTATUS	0x404		This register shows a status that indicates if data sent from the debugger to the CPU has been read.
MAILBOX.TXDATA	0x480		Data sent from the CPU to the debugger.
MAILBOX.TXSTATUS	0x484		This register shows a status that indicates if the data sent from the CPU to the debugger has been read.
ERASEPROTECT.LOCK	0x500		This register locks the ERASEPROTECT.DISABLE register from being written until next reset.
ERASEPROTECT.DISABLE	0x504		This register disables the ERASEPROTECT register and performs an ERASEALL operation.
APPROTECT.LOCK	0x540		This register locks the APPROTECT.DISABLE register from being written to until next reset.

Register	Offset	Security	Description
APPROTECT.DISABLE	0x544		This register disables the APPROTECT register and enables debug access to non-secure mode.
SECUREAPPROTECT.LOCK	0x548		This register locks the SECUREAPPROTECT.DISABLE register from being written until next reset.
SECUREAPPROTECT.DISABLE	0x54C		This register disables the SECUREAPPROTECT register and enables debug access to secure mode.
STATUS	0x600		Status bits for CTRL-AP peripheral.

Table 204: Register overview

8.10.7.1 MAILBOX.RXDATA

Address offset: 0x400

Data sent from the debugger to the CPU.

Reading from this register will automatically set a NoDataPending value in the RXSTATUS 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	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	R	RXDATA			Data received from debugger

8.10.7.2 MAILBOX.RXSTATUS

Address offset: 0x404

This register shows a status that indicates if data sent from the debugger to the CPU has been read.

Bit number	31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value	Description
A	R	RXSTATUS			Status of data in register RXDATA
			NoDataPending	0	No data pending in register RXDATA
			DataPending	1	Data pending in register RXDATA

8.10.7.3 MAILBOX.TXDATA

Address offset: 0x480

Data sent from the CPU to the debugger.

Writing to this register will automatically set a DataPending value in the TXSTATUS 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	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field	Value ID	Value	Description
A	RW	TXDATA			Data sent to debugger

8.10.7.4 MAILBOX.TXSTATUS

Address offset: 0x484

This register shows a status that indicates if the data sent from the CPU to the debugger has been read.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	R	TXSTATUS			Status of data in register TXDATA
			NoDataPending	0	No data pending in register TXDATA
			DataPending	1	Data pending in register TXDATA

8.10.7.5 ERASEPROTECT.LOCK

Address offset: 0x500

This register locks the ERASEPROTECT.DISABLE register from being written until next 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW1	LOCK			Lock ERASEPROTECT.DISABLE register from being written until next reset
			Unlocked	0	Register ERASEPROTECT.DISABLE is writeable
			Locked	1	Register ERASEPROTECT.DISABLE is read-only

8.10.7.6 ERASEPROTECT.DISABLE

Address offset: 0x504

This register disables the ERASEPROTECT register and performs an ERASEALL operation.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW1	KEY			The ERASEALL sequence is initiated if the value of the KEY fields are non-zero and the KEY fields match on both the CPU and debugger sides.

8.10.7.7 APPROTECT.LOCK

Address offset: 0x540

This register locks the APPROTECT.DISABLE register from being written to until next 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW1	LOCK			Lock the APPROTECT.DISABLE register from being written to until next reset
			Unlocked	0	Register APPROTECT.DISABLE is writeable
			Locked	1	Register APPROTECT.DISABLE is read-only

8.10.7.8 APPROTECT.DISABLE

Address offset: 0x544

This register disables the APPROTECT register and enables debug access to non-secure mode.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	KEY			If the value of the KEY field is non-zero, and the KEY fields match on both the CPU and debugger sides, disable APPROTECT and enable debug access to non-secure mode until the next pin reset, brown-out reset, power-on reset, or watchog timer reset. After reset the debugger side register has a fixed KEY value. To enable debug access, both CTRL-AP and UICR.APPROTECT protection needs to be disabled.

8.10.7.9 SECUREAPPROTECT.LOCK

Address offset: 0x548

This register locks the SECUREAPPROTECT.DISABLE register from being written until next 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW1	LOCK			Lock register SECUREAPPROTECT.DISABLE from being written until next reset
			Unlocked	0	Register SECUREAPPROTECT.DISABLE is writeable
			Locked	1	Register SECUREAPPROTECT.DISABLE is read-only

8.10.7.10 SECUREAPPROTECT.DISABLE

Address offset: 0x54C

This register disables the SECUREAPPROTECT register and enables debug access to secure mode.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A 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	R/W	Field	Value ID	Value	Description
A	RW	KEY			If the value of the KEY field is non-zero, and the KEY fields match on both the CPU and debugger sides, disable SECUREAPPROTECT and enable debug access to secure mode until the next pin reset, brown-out reset, power-on reset, or watchog timer reset. After reset the debugger side register has a fixed KEY value. To enable debug access, both CTRL-AP and UICR.SECUREAPPROTECT protection needs to be disabled.

8.10.7.11 STATUS

Address offset: 0x600

Status bits for CTRL-AP peripheral.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	UICRAPPROTECT			Status bit for UICR part of access port protection at last reset. The reset value is automatically read from the APPROTECT register in UICR.
			Enabled	0	APPROTECT was enabled in UICR
			Disabled	1	APPROTECT wasdisabled in UICR
B	R	UICRSECUREAPPROTECT			Status bit for UICR part of secure access port protection at last reset. The reset value is automatically read from the SECUREAPPROTECT register in UICR.
			Enabled	0	SECUREAPPROTECT was enabled in UICR
			Disabled	1	SECUREAPPROTECT was disabled in UICR
C	R	DBGIFACEMODE			Status bit for device debug interface mode
			Disabled	0	No debugger attached
			Enabled	1	Debugger is attached and device is in debug interface mode

8.11 CTI - Cross Trigger Interface

Configuration interface for the Cross Trigger Interface

Please refer to the CTI section for more information about how to configure the Cross Trigger Interface.

8.11.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0xE0042000	APPLICATION	CTI	CTI	S	NA	Cross-trigger interface	Application core CTI
0xE0042000	NETWORK	CTI	CTI	NS	NA	Cross-trigger interface	Network core CTI

Table 205: Instances

Register	Offset	Security	Description
CTICONTROL	0x000		CTI Control register
CTIINTACK	0x010		CTI Interrupt Acknowledge register
CTIAPPSET	0x014		CTI Application Trigger Set register
CTIAPPCLEAR	0x018		CTI Application Trigger Clear register
CTIAPPPULSE	0x01C		CTI Application Pulse register
CTIINEN[n]	0x020		CTI Trigger to Channel Enable register
CTIOUTEN[n]	0x0A0		CTI Channel to Trigger Enable register
CTITRIGINSTATUS	0x130		CTI Trigger In Status register
CTITRIGOUTSTATUS	0x134		CTI Trigger Out Status register
CTICHINSTATUS	0x138		CTI Channel In Status register
CTIGATE	0x140		Enable CTI Channel Gate register
DEVARCH	0xFBC		Device Architecture register
DEVID	0xFC8		Device Configuration register
DEVTYPE	0xFCC		Device Type Identifier register
PIDR4	0xFD0		Peripheral ID4 Register
PIDR5	0xFD4		Peripheral ID5 register
PIDR6	0xFD8		Peripheral ID6 register
PIDR7	0xFDC		Peripheral ID7 register
PIDR0	0xFE0		Peripheral ID0 Register
PIDR1	0xFE4		Peripheral ID1 Register
PIDR2	0xFE8		Peripheral ID2 Register
PIDR3	0xFEC		Peripheral ID3 Register
CIDR0	0xFF0		Component ID0 Register
CIDR1	0xFF4		Component ID1 Register
CIDR2	0xFF8		Component ID2 Register
CIDR3	0xFFC		Component ID3 Register

Table 206: Register overview

8.11.1.1 CTICONTROL

Address offset: 0x000

CTI Control register

The CTICONTROL register enables the CTI.

Bit number	31	30	29	28	27	26	25	24	23	22	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	R/W	Field	Value ID	Value	Description
A	RW	GLBEN			Enables or disables the CTI.
			Disabled	0	All cross-triggering mapping logic functionality is disabled.
			Enabled	1	Cross-triggering mapping logic functionality is enabled.

8.11.1.2 CTIINTACK

Address offset: 0x010

CTI Interrupt Acknowledge register

The CTIINTACK register is a software acknowledge for a trigger output. This register is used when ctitrigout is used as a sticky output. That is, no hardware acknowledge is available and a software acknowledge is required.

Bit number						31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																														H	G	F	E	D	C	B	A
Reset 0x00000000						**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field		Value ID		Value								Description																							
A-H	W	INTACK[i] (i=0..7)												Acknowledges the ctitrigout i output.																							
				Acknowledge		1								Clears the ctitrigout.																							

8.11.1.3 CTIAPPSET

Address offset: 0x014

CTI Application Trigger Set register

Writing to the CTIAPPSET register causes a channel event to be raised, corresponding to the bit written to.

Bit number						31	30	29	28	27	26	25	24	23	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	R/W	Field		Value ID		Value								Description																							
A-D	RW	APPSET[i] (i=0..3)												Application trigger event for channel i.																							
				Inactive		0								Application trigger i is inactive.																							
				Active		1								Application trigger i is active.																							
				Activate		1								Generate channel event for channel i.																							

8.11.1.4 CTIAPPCLEAR

Address offset: 0x018

CTI Application Trigger Clear register

Writing to a bit in the CTIAPPCLEAR register clears the corresponding channel 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	B	A
Reset 0x00000000						**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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/W	Field		Value ID		Value								Description																							
A-D	W	APPCLEAR[i] (i=0..3)												Sets the corresponding bits in the CTIAPPSET to 0. There is one bit of the register for each channel.																							
				Clear		1								Clears the event for channel i.																							

8.11.1.5 CTIAPPPULSE

Address offset: 0x01C

CTI Application Pulse register

A write to this register causes a channel event pulse of one cticlk period to be generated. This corresponds to the bit that was written to. The pulse external to the CTI can be extended to multi-cycle by the

handshaking interface circuits. This register clears itself immediately, so it can be repeatedly written to without software having to clear it.

Bit number					31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A-D	W	APPULSE[i] (i=0..3)			Setting a bit HIGH generates a channel event pulse for the selected channel. There is one bit of the register for each channel.
			Generate	1	Generates an event pulse on channel i.

8.11.1.6 CTIINEN[n] (n=0..7)

Address offset: 0x020 + (n × 0x4)

CTI Trigger to Channel Enable register

The CTIINENn register enables the signaling of an event on CTM channels when a trigger event is received by the CTI. There is a bit for each of the four channels implemented. This register does not affect the application trigger operations.

Bit number					31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A-D	RW	TRIGINEN[i] (i=0..3)			Enables a cross trigger event to channel i when a ctitrigin input is activated.
			Disabled	0	Input trigger n events are ignored by channel i.
			Enabled	1	When an event is received on input trigger n (ctitrigin[n]), generate an event on channel i.

8.11.1.7 CTIOUTEN[n] (n=0..7)

Address offset: 0x0A0 + (n × 0x4)

CTI Channel to Trigger Enable register

The CTIOUTENn register defines which channels can generate a ctitrigout[n] output. There is a bit for each of the four channels implemented. This register affects the mapping from application trigger to trigger outputs.

Bit number					31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A-D	RW	TRIGOUTEN[i] (i=0..3)			Enables a cross trigger event to ctitrigout when channel i is activated.
			Disabled	0	Channel i is ignored by output trigger n.
			Enabled	1	When an event occurs on channel i, generate an event on output event n (ctitrigout[n]).

8.11.1.8 CTITRIGINSTATUS

Address offset: 0x130

CTI Trigger In Status register

The CTITRIGINSTATUS register provides the status of the ctitrigin inputs.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	R	TRIGINSTATUS[i] (i=0..7)			Shows the status of ctitrigini input.
			Active	1	Ctitrigin i is active.
			Inactive	0	Ctitrigin i is inactive.

8.11.1.9 CTITRIGOUTSTATUS

Address offset: 0x134

CTI Trigger Out Status register

The CTITRIGOUTSTATUS register provides the status of the ctitrigout outputs.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					H G F E D C B A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A-H	R	TRIGOUTSTATUS[i] (i=0..7)			Shows the status of ctitrigouti output.
			Active	1	Ctitrigout i is active.
			Inactive	0	Ctitrigout i is inactive.

8.11.1.10 CTICHINSTATUS

Address offset: 0x138

CTI Channel In Status register

The CTICHINSTATUS register provides the status of the ctichin inputs.

Bit number					31 30 29 28 27 26 25 24 23 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	R/W	Field	Value ID	Value	Description
A-D	R	CTICHINSTATUS[i] (i=0..3)			Shows the status of the ctitrigin i input.
			Active	1	Ctichin i is active.
			Inactive	0	Ctichin i is inactive.

8.11.1.11 CTIGATE

Address offset: 0x140

Enable CTI Channel Gate register

The CTIGATE register prevents the channels from propagating through the CTM to other CTIs. This enables local cross-triggering (e.g. causing an interrupt when the ETM trigger occurs). It can be used effectively with CTIAPPSET, CTIAPPCLEAR, and CTIAPPPULSE for asserting trigger outputs by asserting channels, without affecting the rest of the system. On reset, this register is 0xF and channel propagation is enabled.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x0000000F				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A-D	RW	CTIGATEEN[i] (i=0..3)			Enable ctichouti.
			Enabled	1	Enable ctichout channel i propagation.
			Disabled	0	Disable ctichout channel i propagation.

8.11.1.12 DEVARCH

Address offset: 0xFBC

Device Architecture 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 0x47701A14				**0 1 0 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	Architecture			Contains the CTI device architecture.

8.11.1.13 DEVID

Address offset: 0xFC8

Device Configuration register

The DEVID register indicates the capabilities of the component.

Bit number				31 30 29 28 27 26 25 24 23 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 B B B B B B B B A A A A A
Reset 0x00040800				**0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	EXTMUXNUM			Indicates the number of multiplexers available on Trigger Inputs and Trigger Outputs that are using asicctl. The default value of 0b00000 indicates that no multiplexing is present.
B	R	NUMTRIG			Number of ECT triggers available.
C	R	NUMCH			Number of ECT channels available.

8.11.1.14 DEVTYPE

Address offset: 0xFCC

Device Type Identifier register

The DEVTYPE register provides a debugger with information about the component when the Part Number field is not recognized. The debugger can then report this information.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B A A A A
Reset 0x00000014				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	MAJOR			Major classification of the type of the debug component as specified in the Arm Architecture Specification for this debug and trace component.
			Controller	4	Indicates that this component allows a debugger to control other components in an Arm CoreSight SoC-400 system.
B	R	SUB			Sub-classification of the type of the debug component as specified in the Arm Architecture Specification within the major classification as specified in the MAJOR field.
			Crosstrigger	1	Indicates that this component is a sub-triggering component.

8.11.1.15 PIDR4

Address offset: 0xFD0

Peripheral ID4 Register

The PIDR4 register is part of the set of peripheral identification registers. It contains part of the designer identity and the memory size.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B A A A A
Reset 0x00000004				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0**
ID	R/W	Field	Value ID	Value	Description
A	R	DES_2			Together, PIDR1.DES_0, PIDR2.DES_1, and PIDR4.DES_2 identify the designer of the component.
			Code	4	JEDEC continuation code.
B	R	SIZE			Always 0b0000. Indicates that the device only occupies 4KB of memory.

8.11.1.16 PIDR5

Address offset: 0xFD4

Peripheral ID5 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**
ID	R/W	Field	Value ID	Value	Description

8.11.1.17 PIDR6

Address offset: 0xFD8

Peripheral ID6 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**
ID	R/W	Field	Value ID	Value	Description

8.11.1.18 PIDR7

Address offset: 0xFDC

Peripheral ID7 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**
ID	R/W	Field	Value ID	Value	Description

8.11.1.19 PIDR0

Address offset: 0xFE0

Peripheral ID0 Register

The PIDR0 register is part of the set of peripheral identification registers. It contains part of the designer-specific part 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 0x00000021				**0 0 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 1**
ID	R/W	Field	Value ID	Value	Description
A	R	PART_0			Bits[7:0] of the 12-bit part number of the component. The designer of the component assigns this part number.
			PartnumberL	0x21	Indicates bits[7:0] of the part number of the component.

8.11.1.20 PIDR1

Address offset: 0xFE4

Peripheral ID1 Register

The PIDR1 register is part of the set of peripheral identification registers. It contains part of the designer-specific part number and part of the designer identity.

Bit number				31 30 29 28 27 26 25 24 23 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 A A A
Reset 0x000000BD				**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 1 0 1**
ID	R/W	Field	Value ID	Value	Description
A	R	PART_1			Bits[11:8] of the 12-bit part number of the component. The designer of the component assigns this part number.
			PartnumberH	13	Indicates bits[11:8] of the part number of the component.
B	R	DES_0			Together, PIDR1.DES_0, PIDR2.DES_1, and PIDR4.DES_2 identify the designer of the component.
			Arm	11	Arm. Bits[3:0] of the JEDEC JEP106 Identity Code

8.11.1.21 PIDR2

Address offset: 0xFE8

Peripheral ID2 Register

The PIDR2 register is part of the set of peripheral identification registers. It contains part of the designer identity and the product revision.

Bit number				31 30 29 28 27 26 25 24 23 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 B A A A
Reset 0x0000000B				**0 0 0 0 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**
ID	R/W	Field	Value ID	Value	Description
A	R	DES_1			Together, PIDR1.DES_0, PIDR2.DES_1, and PIDR4.DES_2 identify the designer of the component.
			Arm	3	Arm. Bits[6:4] of the JEDEC JEP106 Identity Code
B	R	JEDEC			Always 1. Indicates that the JEDEC-assigned designer ID is used.
C	R	REVISION			Peripheral revision
			Rev0p0	0	This device is at r0p0

8.11.1.22 PIDR3

Address offset: 0xFEC

Peripheral ID3 Register

The PIDR3 register is part of the set of peripheral identification registers. It contains the REVAND and CMOD fields.

Bit number				31 30 29 28 27 26 25 24 23 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 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	R/W	Field	Value ID	Value	Description
A	R	CMOD			Customer Modified. Indicates whether the customer has modified the behavior of the component. In most cases, this field is 0b0000. Customers change this value when they make authorized modifications to this component.
			Unmodified	0	Indicates that the customer has not modified this component.
B	R	REVAND			Indicates minor errata fixes specific to the revision of the component being used, for example metal fixes after implementation. In most cases, this field is 0b0000. Arm recommends that the component designers ensure that a metal fix can change this field if required, for example, by driving it from registers that reset to 0b0000.
			NoErrata	0	Indicates that there are no errata fixes to this component.

8.11.1.23 CIDR0

Address offset: 0xFF0

Component ID0 Register

The CIDR0 register is a component identification register that indicates the presence of identification registers.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x0000000D				**0 0 0 0 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**
ID	R/W	Field	Value ID	Value	Description
A	R	PRMBL_0			Preamble[0]. Contains bits[7:0] of the component identification code.
			Value	0x0D	Bits[7:0] of the identification code.

8.11.1.24 CIDR1

Address offset: 0xFF4

Component ID1 Register

The CIDR1 register is a component identification register that indicates the presence of identification registers. This register also indicates the component class.

Bit number				31 30 29 28 27 26 25 24 23 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 A A A
Reset 0x00000090				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	R	PRMBL_1			Preamble[1]. Contains bits[11:8] of the component identification code.
			Value	0	Bits[11:8] of the identification code.
B	R	CLASS			Class of the component, for example, whether the component is a ROM table or a generic CoreSight component. Contains bits[15:12] of the component identification code
			Coresight	9	Indicates that the component is a CoreSight component.

8.11.1.25 CIDR2

Address offset: 0xFF8

Component ID2 Register

The CIDR2 register is a component identification register that indicates the presence of identification registers.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x00000005				**0 0 0 0 0 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**

ID	R/W	Field	Value ID	Value	Description
A	R	PRMBL_2			Preamble[2]. Contains bits[23:16] of the component identification code.
			Value	0x05	Bits[23:16] of the identification code.

8.11.1.26 CIDR3

Address offset: 0xFFC

Component ID3 Register

The CIDR3 register is a component identification register that indicates the presence of identification registers.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A
Reset 0x000000B1				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1**

ID	R/W	Field	Value ID	Value	Description
A	R	PRMBL_3			Preamble[3]. Contains bits[31:24] of the component identification code.
			Value	0xB1	Bits[31:24] of the identification code.

8.12 TAD - Trace and debug control

Configuration interface for trace and debug

Please refer to the Trace section for more information about how to configure the trace and debug interface.

Note: Although there are PSEL registers for the trace port, each function can only be mapped to a single pin due to pin speed requirements. Setting the PIN field to anything else will not have any effect. See Pin assignment chapter for more information

8.12.1 Registers

Base address	Domain	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0xE0080000	APPLICATION	TAD	TAD	S	NA	Trace and debug control	

Table 207: Instances

Register	Offset	Security	Description	
CLOCKSTART	0x004		Start all trace and debug clocks.	
CLOCKSTOP	0x008		Stop all trace and debug clocks.	
ENABLE	0x500		Enable debug domain and aquire selected GPIOs	
PSEL.TRACECLK	0x504		Pin configuration for TRACECLK	
PSEL.TRACEDATA0	0x508		Pin configuration for TRACEDATA[0]	
PSEL.TRACEDATA1	0x50C		Pin configuration for TRACEDATA[1]	
PSEL.TRACEDATA2	0x510		Pin configuration for TRACEDATA[2]	
PSEL.TRACEDATA3	0x514		Pin configuration for TRACEDATA[3]	
TRACEPORTSPEED	0x518		Clocking options for the Trace Port debug interface Reset behavior is the same as debug components	Retained

Table 208: Register overview

8.12.1.1 CLOCKSTART

Address offset: 0x004

Start all trace and debug clocks.

Bit number	31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	START			
			Start	1	Start all trace and debug clocks.

8.12.1.2 CLOCKSTOP

Address offset: 0x008

Stop all trace and debug clocks.

Bit number				31 30 29 28 27 26 25 24 23 22 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	R/W	Field	Value ID	Value	Description
A	W	STOP			
			Stop	1	Stop all trace and debug clocks.

8.12.1.3 ENABLE

Address offset: 0x500

Enable debug domain and aquire selected 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	ENABLE			
			DISABLED	0	Disable debug domain and release selected GPIOs
			ENABLED	1	Enable debug domain and aquire selected GPIOs

8.12.1.4 PSEL.TRACECLK

Address offset: 0x504

Pin configuration for TRACECLK

Bit number				31 30 29 28 27 26 25 24 23 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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN			Pin number
			Traceclk	12	TRACECLK pin
					Note: Only this pin is valid
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.12.1.5 PSEL.TRACEDATA0

Address offset: 0x508

Pin configuration for TRACEDATA[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				B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN			Pin number
			Tracedata0	11	TRACEDATA0 pin
					Note: Only this pin is valid
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.12.1.6 PSEL.TRACEDATA1

Address offset: 0x50C

Pin configuration for TRACEDATA[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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN			Pin number
			Tracedata1	10	TRACEDATA1 pin
					Note: Only this pin is valid
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.12.1.7 PSEL.TRACEDATA2

Address offset: 0x510

Pin configuration for TRACEDATA[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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN			Pin number
			Tracedata2	9	TRACEDATA2 pin
					Note: Only this pin is valid
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.12.1.8 PSEL.TRACEDATA3

Address offset: 0x514

Pin configuration for TRACEDATA[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
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN			Pin number
			Tracedata3	8	TRACEDATA3 pin
					Note: Only this pin is valid
B	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.12.1.9 TRACEPORTSPEED (Retained)

Address offset: 0x518

This register is a retained register

Clocking options for the Trace Port debug interface

Reset behavior is the same as debug components

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 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/W	Field	Value ID	Value	Description
A	RW	TRACEPORTSPEED			Speed of Trace Port clock. Note that the TRACECLK pin output will be divided again by two from the Trace Port clock.
			64MHz	0	Trace Port clock is: 64MHz
			32MHz	1	Trace Port clock is: 32MHz
			16MHz	2	Trace Port clock is: 16MHz
			8MHz	3	Trace Port clock is: 8MHz

9 Hardware and layout

9.1 Pin assignments

This section describes the pin assignment and the pin functions.

This device provides flexibility when it comes to routing and configuration of the GPIO pins. However, some pins have recommendations for how the pin should be configured or what it should be used for.

In addition to the information in the pinout tables for the respective packages, the following peripherals have dedicated pins that should be used for proper operation:

- TWI - For the fastest TWI 1 Mbps mode, the two high-speed TWI pins must be configured in the TWI peripheral's PSEL registers, and the 20 mA open drain driver enabled using the E0E1 drive setting in the DRIVE field of the PIN_CNF GPIO register.
- QSPI - For QSPI only the dedicated GPIO pins from the following table shall be used. These must be enabled using the Peripheral option of the PIN_CNF[p].MCUSEL register. The GPIO must use the high drive H0H1 configuration in the DRIVE field of the PIN_CNF GPIO register.
- SPIM4 - For the fastest SPI mode, the special purpose GPIO pins are enabled using the Peripheral option of the PIN_CNF[p].MCUSEL register. When activated, the SPIM PSEL settings are ignored, and the dedicated pins are used. The GPIO must use the high drive H0H1 configuration in the DRIVE field of the PIN_CNF GPIO register.
- TRACE - When using trace, the TRACEDATA[n] and TRACECLK GPIO pins must all use the extra high drive E0E1 configuration in the DRIVE field of the PIN_CNF GPIO register. Also, the TND option of the PIN_CNF[p].MCUSEL register must be used.

GPIO pin	Description
P0.08 - P0.12	Drive configuration E0E1 is available and must be used for TRACE. For 32 Mbps high-speed SPI using SPIM4, drive configuration H0H1 must be used.
P0.13 - P0.18	The H0H1 drive configuration features the highest speeds of quad SPI using the direct connection of the QSPI peripheral.
P1.02 and P1.03	The E0E1 drive configuration activates a 20 mA open-drain driver specifically designed for high-speed TWI.
Remaining pins	The E0E1 drive configuration is not supported. Using the E0E1 drive configuration will cause incorrect operation.

Table 209: Special GPIO considerations

Note: The extra high drive E0E1 drive configuration has limited availability. It is only available for the dedicated TRACE pins on P0.08 through P0.12. For the dedicated, high-speed TWIM pins on P1.02 and P1.03, the E0E1 drive configuration activates a powerful 20 mA *open-drain* driver specifically designed for high-speed TWI.

For all high-speed signals, the printed circuit board (PCB) layout must ensure that connections are made using short PCB traces. Refer to the manufacturer's PCB design recommendations for additional information.

9.1.1 aQFN94 pin assignments

The aQFN94 package has 94 pins in addition to four corner pads and a die pad.

Figure 252: aQFN pin assignments, top view. Corner and die pad is not illustrated.

Pin	Name	Function	Description	Recommended usage
A5	VBUS	Power	5 V input for USB 3.3 V regulator	
A13	DECA	Power	Analog regulator supply decoupling	
A15	DECD	Power	Digital regulator supply decoupling	
A17	P1.13	Digital I/O	General purpose I/O	
A19	VDD	Power	Power supply	
A21	DCC	Power	DC/DC converter output	
A23	DECN	Power	Regulator supply decoupling	
A25	N.C.			
A27	DECR	Power	Regulator supply decoupling	
B2	D+	USB	USB D+	
B4	D-	USB	USB D-	
B6	DECUSB	Power	USB 3.3 V regulator supply decoupling	
B8	VDD	Power	Power supply	
B10	DCCD	Power	DC/DC converter output	
B12	N.C.			
B14	P1.15	Digital I/O	General purpose I/O	
B16	P1.14	Digital I/O	General purpose I/O	
B18	P1.12	Digital I/O	General purpose I/O	
B20	P1.11	Digital I/O	General purpose I/O	
B22	P0.31	Digital I/O	General purpose I/O	
B24	P0.30	Digital I/O	General purpose I/O	
B26	N.C.			
B28	VDD	Power	Power supply	
B30	XC2	Analog input	Connection for 32 MHz crystal	
C1	VDD	Power	Power supply	
C31	XC1	Analog input	Connection for 32 MHz crystal	
D2	N.C.			
E1	VDDH	Power	Power supply	
E31	VDD	Power	Power supply	
F2	N.C.			
G1	N.C.			
G31	DECRF	Power	RADIO power supply decoupling	
H2	N.C.			
J1	DCCH	Power	DC/DC converter output	
J31	N.C.			
K2	N.C.			
L1	VDD	Power	Power supply	
L31	ANT	RF	Single-ended antenna connection	
M2	P1.00	Digital I/O	General purpose I/O	
N1	P0.00	Digital I/O	General purpose I/O	
	XL1	Analog input	Connection for 32 kHz crystal	
N31	VDD	Power	Power supply	
P2	P1.01	Digital I/O	General purpose I/O	
R1	P0.01	Digital I/O	General purpose I/O	
	XL2	Analog input	Connection for 32 kHz crystal	
R31	P1.10	Digital I/O	General purpose I/O	
T2	N.C.			
U1	VDD	Power	Power supply	
U31	P0.29	Digital I/O	General purpose I/O	
V2	P0.04	Digital I/O	General purpose I/O	
	AIN0	Analog input	Analog input	
W1	P0.02	Digital I/O	General purpose I/O	
	NFC1	NFC input	NFC antenna connection	

Pin	Name	Function	Description	Recommended usage
W31	SWDCLK	Debug	Serial wire debug clock input for debug and programming	
Y2	P0.05	Digital I/O	General purpose I/O	
	AIN1	Analog input	Analog input	
AA1	P0.03	Digital I/O	General purpose I/O	
	NFC2	NFC input	NFC antenna connection	
AA31	SWDIO	Debug	Serial wire debug I/O for debug and programming	
AB2	P0.06	Digital I/O	General purpose I/O	
	AIN2	Analog input	Analog input	
AC1	VDD	Power	Power supply	
AC31	nRESET	Reset	Pin RESET with internal pull-up resistor	
AD2	P0.07	Digital I/O	General purpose I/O	
	AIN3	Analog input	Analog input	
AE1	P1.02	Digital I/O	General purpose I/O	TWI
	TWI	TWI 1 Mbps	High-speed pin for 1 Mbps TWI	
AE31	P0.28	Digital I/O	General purpose I/O	
	AIN7	Analog input	Analog input	
AF2	P1.03	Digital I/O	General purpose I/O	TWI
	TWI	TWI 1 Mbps	High-speed pin for 1 Mbps TWI	
AG1	VDD	Power	Power supply	
AG31	N.C.			
AH2	P0.08	Digital I/O	General purpose I/O	Trace, SPIM4
	TRACEDATA3	Trace data	Trace buffer TRACEDATA[3]	
	SCK	SCK for SPIM4	Dedicated pin for high-speed SPI	
AJ1	P0.09	Digital I/O	General purpose I/O	Trace, SPIM4
	TRACEDATA2	Trace data	Trace buffer TRACEDATA[2]	
	MOSI	MOSI for SPIM4	Dedicated pin for high-speed SPI	
AJ31	VDD	Power	Power supply	
AK2	P0.10	Digital I/O	General purpose I/O	Trace, SPIM4
	TRACEDATA1	Trace data	Trace buffer TRACEDATA[1]	
	MISO	MISO for SPIM4	Dedicated pin for high-speed SPI	
AK4	P0.11	Digital I/O	General purpose I/O	Trace, SPIM4
	TRACEDATA0	Trace data	Trace buffer TRACEDATA[0]	
	CSN	CSN for SPIM4	Dedicated pin for high-speed SPI	
AK6	P0.12	Digital I/O	General purpose I/O	Trace, SPIM4
	TRACECLK	Trace clock	Trace buffer clock	
	DCX	DCX for SPIM4	Dedicated pin for high-speed SPI	
AK8	P0.14	Digital I/O	General purpose I/O	QSPI
	IO1	IO1 for QSPI	Dedicated pin for Quad SPI	
AK10	P0.15	Digital I/O	General purpose I/O	QSPI
	IO2	IO2 for QSPI	Dedicated pin for Quad SPI	
AK12	P0.17	Digital I/O	General purpose I/O	QSPI
	SCK	SCK for QSPI	Dedicated pin for Quad SPI	
AK14	P0.18	Digital I/O	General purpose I/O	QSPI
	CSN	CSN for QSPI	Dedicated pin for Quad SPI	
AK16	P0.20	Digital I/O	General purpose I/O	
AK18	P0.22	Digital I/O	General purpose I/O	
AK20	P0.23	Digital I/O	General purpose I/O	
AK22	P1.05	Digital I/O	General purpose I/O	
AK24	P1.07	Digital I/O	General purpose I/O	
AK26	P1.09	Digital I/O	General purpose I/O	
AK28	P0.25	Digital I/O	General purpose I/O	
	AIN4	Analog input	Analog input	
AK30	P0.27	Digital I/O	General purpose I/O	

Pin	Name	Function	Description	Recommended usage
	AIN6	Analog input	Analog input	
AL3	VDD	Power	Power supply	
AL5	P0.13	Digital I/O	General purpose I/O	QSPI
	IO0	IO0 for QSPI	Dedicated pin for Quad SPI	
AL7	VDD	Power	Power supply	
AL9	P0.16	Digital I/O	General purpose I/O	QSPI
	IO3	IO3 for QSPI	Dedicated pin for Quad SPI	
AL11	VDD	Power	Power supply	
AL13	P0.19	Digital I/O	General purpose I/O	
AL15	P0.21	Digital I/O	General purpose I/O	
AL17	VDD	Power	Power supply	
AL19	P1.04	Digital I/O	General purpose I/O	
AL21	P1.06	Digital I/O	General purpose I/O	
AL23	P1.08	Digital I/O	General purpose I/O	
AL25	VDD	Power	Power supply	
AL27	P0.24	Digital I/O	General purpose I/O	
AL29	P0.26	Digital I/O	General purpose I/O	
	AIN5	Analog input	Analog input	
Corner pads				
A1	N.C.			
A31	N.C.			
AL1	N.C.			
AL31	N.C.			
Bottom of chip				
Die pad	VSS	Power	Ground pad. Exposed die pad must be connected to ground (VSS) for proper device operation.	

Table 210: aQFN pin assignments

9.2 Mechanical specifications

The mechanical specifications for the packages show the dimensions in millimeters.

9.2.1 aQFN94 7 x 7 mm package

Dimensions in millimeters for the aQFN94 7 x 7 mm package.

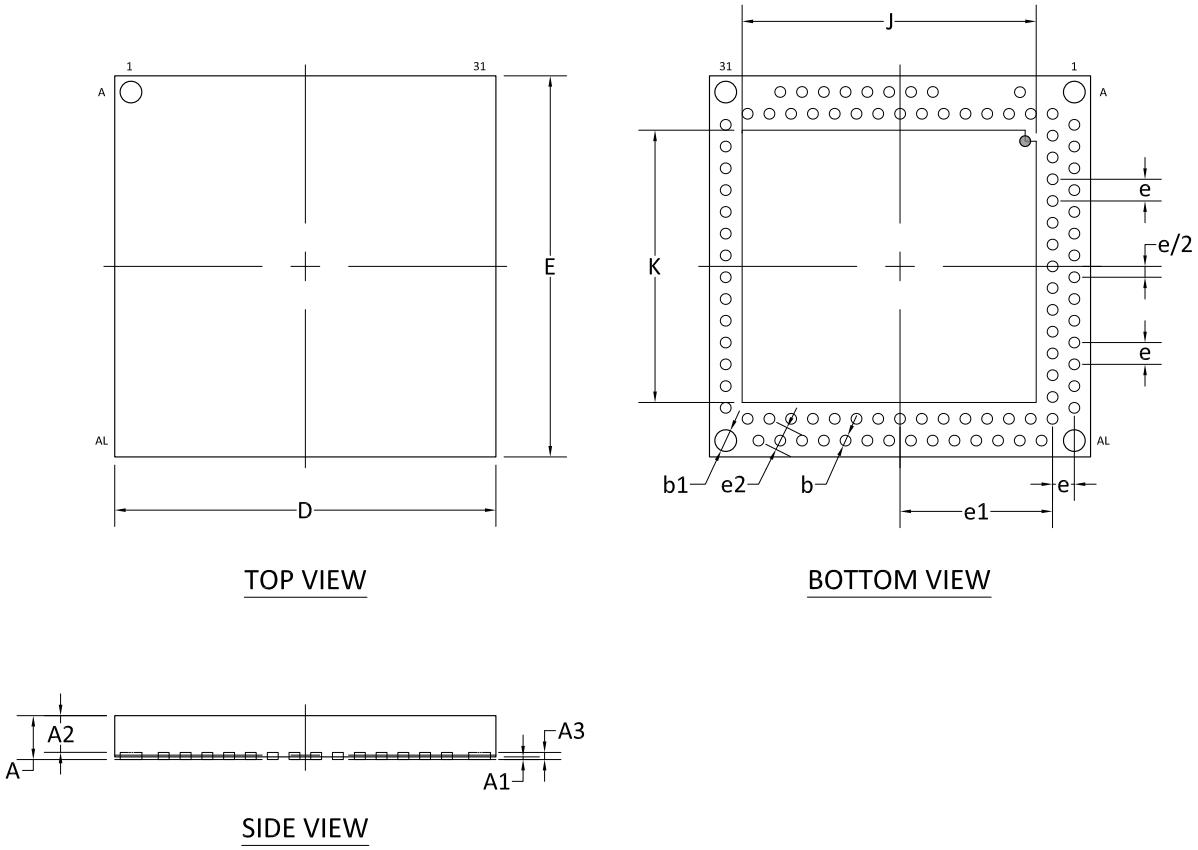

Figure 253: aQFN94 7 x 7 mm package

	A	A1	A2	A3	b	b1	D, E	e	e1	e2	J	K
Min.		0.02			0.15						5.3	4.9
Nom.		0.05	0.675	0.13	0.20	0.4	7.00	0.4	2.8	0.447	5.4	5.0
Max.	0.85	0.08			0.25						5.5	5.1

Table 211: aQFN94 dimensions in millimeters

9.3 Reference circuitry

To ensure good RF performance when designing PCBs, it is highly recommended to use the PCB layouts and component values provided by Nordic Semiconductor.

Documentation for the different package reference circuits, including Altium Designer files, PCB layout files, and PCB production files can be downloaded from www.nordicsemi.com.

In this section, there is a reference circuit for QKAA aQFN94, showing the components and component values to support on-chip features in a design.

Some general guidance is summarized here:

- When supplying power from a USB source only, VBUS must be connected to VDDH if USB is to be used.
- Components required for DC/DC function are only needed if DC/DC mode is enabled for that regulator.
- NFC can be used in any configuration.
- USB can be used in any configuration as long as VBUS is supplied by the USB host.

- The schematics include an optional, but recommended, series resistor on the USB supply for improved immunity to transient over-voltage during VBUS connection.

9.3.1 Circuit configuration no. 1 for QKAA aQFN94

Circuit configuration number 1 for QKAA aQFN94 is showing the schematic and the bill of materials table.

Config no.	Supply configuration		Enabled features	
	VDDH	VDD	USB	NFC
Config. 1	USB	N/A	Yes	No

Table 212: Configuration summary for circuit configuration no. 1

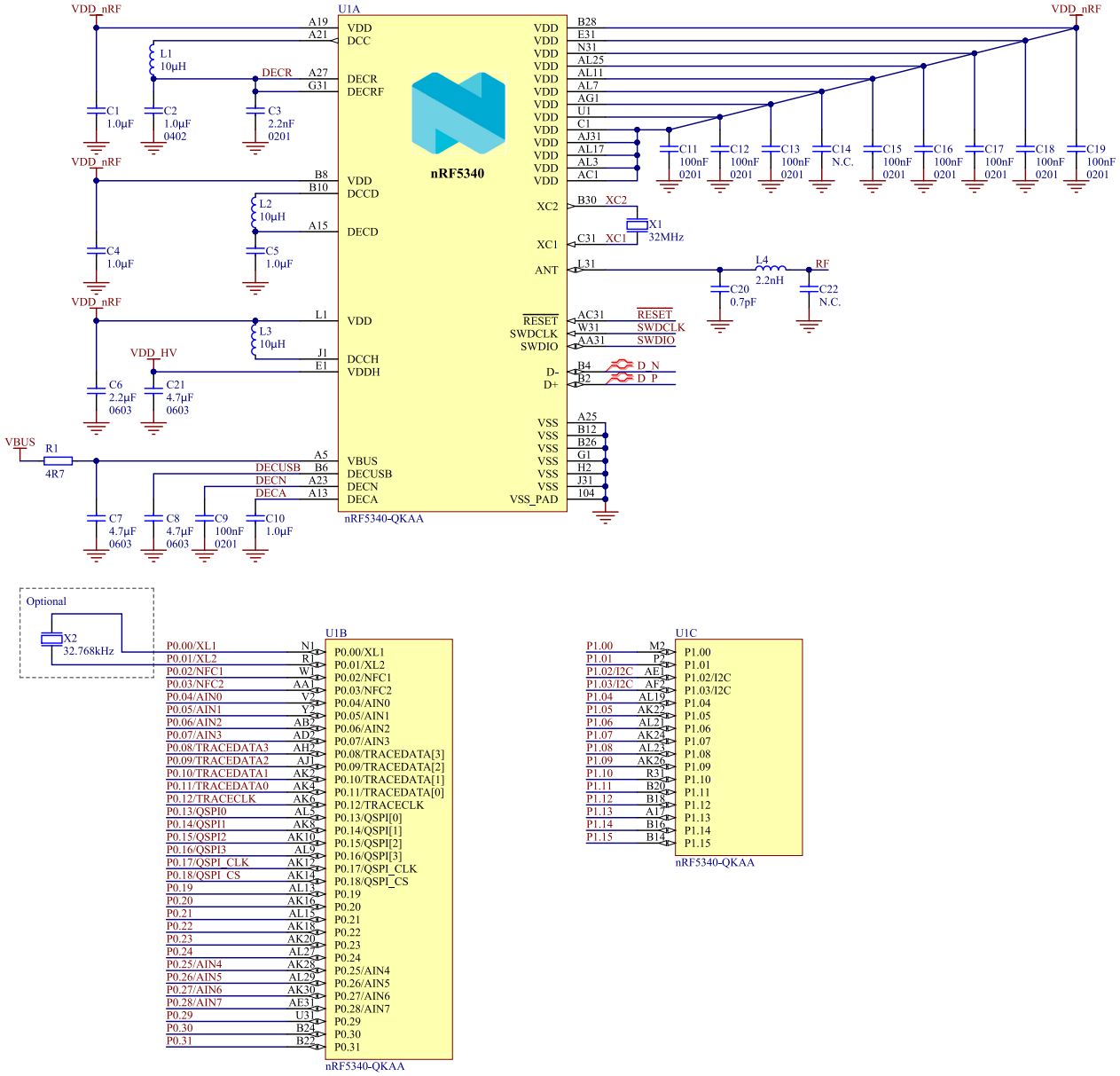

Figure 254: Circuit configuration no. 1 schematic

Note: For PCB reference layouts, see the product page for the nRF5340 on www.nordicsemi.com.

Designator	Value	Description	Footprint
C1, C2, C4, C5, C10	1.0 µF	Capacitor, X5R, ±5%, 6.3 V	0402
C3	2.2 nF	Capacitor, X7R, ±10%, 10V	0201
C6	2.2 µF	Capacitor, X7R, ±10%	0603
C7	4.7 µF	Capacitor, X7R, ±10%	0603
C8, C21	4.7 µF	Capacitor, X7S, ±10%	0603
C9, C11, C12, C13, C15, C16, C17, C18, C19	100 nF	Capacitor, X5R, ±10%	0201
C20	0.7 pF	Capacitor, NP0, ±0.05 pF	0201
C14, C22	N.C.	Not mounted	0201
L1, L2, L3	10 µH	Inductor, 80 mA, ±10%	0603
L4	2.2 nH	High frequency chip inductor, ±5%	0201
R1	4.7 Ω	Resistor, ±1%, 0.05 W	0201
U1	nRF5340-QKAA	Multiprotocol Bluetooth Low Energy, IEEE 802.15.4, ANT, and 2.4GHz proprietary System on Chip	AQFN-94
X1	32 MHz	Crystal SMD 2016, 32 MHz, Cl=8 pF, Total Tol: ±40 ppm	XTAL_2016
X2	32.768 kHz	Crystal SMD 2012, 32.768 kHz, Cl=9 pF, Total Tol: ±50 ppm	XTAL_2012

Table 213: Bill of material for circuit configuration no. 1

9.3.2 PCB layout example

The PCB layout in the following figure is a reference layout for the aQFN package with internal LDO setup and VBUS supply.

Note: C12 is grounded via pin J31 and to the VSS die pad. This is done to create additional filtering of harmonic components.

For all available reference layouts, see the product page for nRF5340 on www.nordicsemi.com.

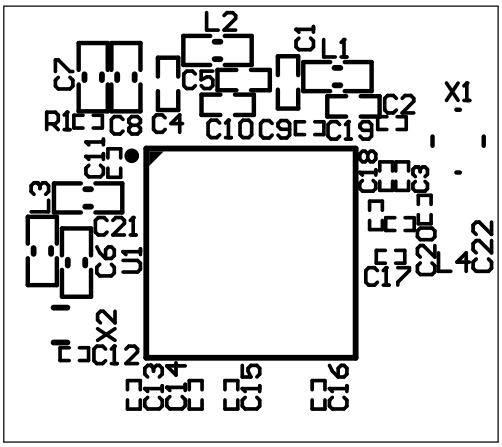

Figure 255: Top silk layer

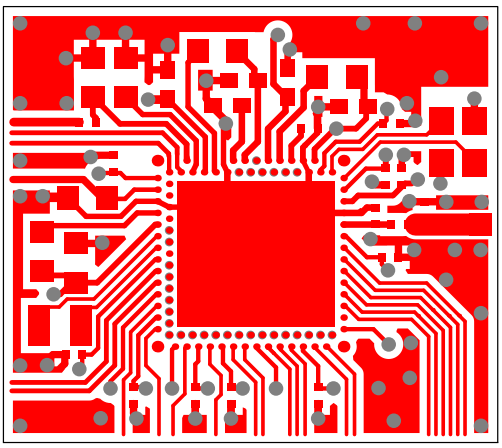

Figure 256: Top layer

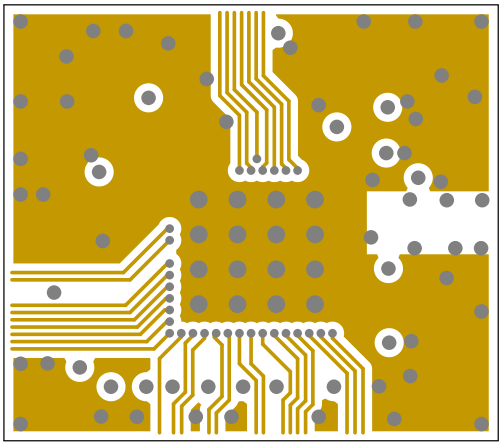

Figure 257: Mid layer 1

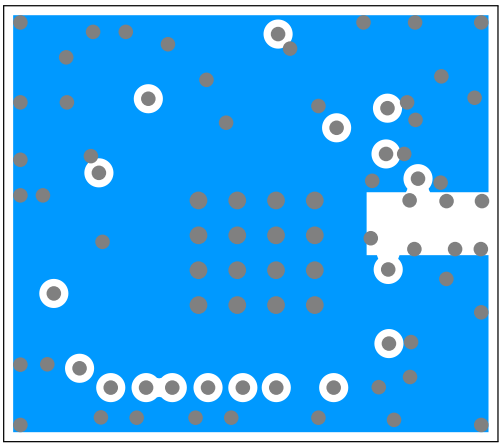

Figure 258: Mid layer 2

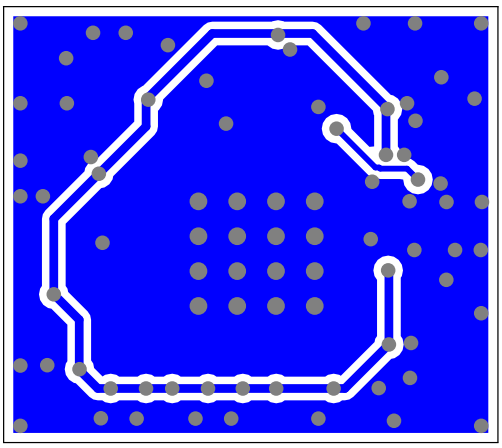

Figure 259: Bottom layer

10 Recommended operating conditions

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

Symbol	Parameter	Min.	Nom.	Max.	Units
VDD	VDD supply voltage, independent of DCDC enable	1.7	3.0	3.6	V
VDDH	VDDH supply voltage, independent of DCDC enable	2.5	3.7	5.5	V
VBUS	VBUS USB supply voltage	4.35	5.0	5.5	V
TA	Operating temperature	-40	25	105	°C

Table 214: Recommended operating conditions

11 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[33].

	Min.	Max.	Unit
Supply voltages			
VDD	-0.3	+3.9	V
VDDH	-0.3	+5.8	V
VBUS	-0.3	+5.8	V
VSS		0	V
I/O pin voltage			
$V_{I/O}$, VDD ≤ 3.6 V	-0.3	VDD + 0.3	V
$V_{I/O}$, VDD > 3.6 V	-0.3	3.9	V
Environmental aQFN package			
Storage temperature	-40	+125	°C
Moisture Sensitivity Level (MSL)		2	
ESD Human Body Model (HBM)		2 (all pins except DECR and DECN, rated at 1.4 kV)	kV
ESD Charged Device Model (CDM)		500	V
Flash memory			
Endurance	10 000 write/erase cycles		
Retention	10 years at 40°C		

Table 215: Absolute maximum ratings

[33] For accellerated life time testing (HTOL, etc) supply voltage should not exceed the recommended operating conditions max value, see Recommended operating conditions on page 793.

12 Ordering information

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

12.1 Device marking

The nRF5340 package is marked as shown in the following figure.

N	5	3	4	0	
<P	P>	<V	V>	<H>	<P>
<Y	Y>	<W	W>	<L	L>

Figure 260: Device marking

12.2 Box labels

The following figures show the box labels used for nRF5340.

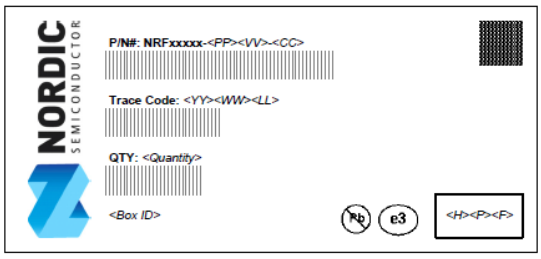

Figure 261: Inner box label

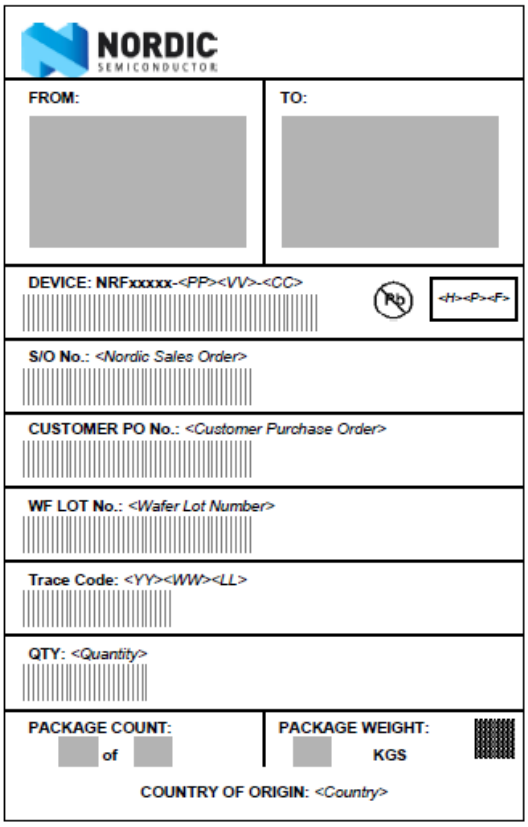

Figure 262: Outer box label

12.3 Order code

The following are the order codes and definitions for nRF5340.

n	R	F	5	3	4	0	-	<P	P>	<V	V>	-	<C	C>

Figure 263: Order code

Abbreviation	Definition and implemented codes
N53/nRF53	nRF53 series product
40	Part code
<PP>	Package variant code
<VV>	Function variant code
<H><P><F>	Build code H - Hardware version code P - Production configuration code (production site, etc.) F - Firmware version code (only visible on shipping container label)
<YY><WW><LL>	Tracking code YY - Year code WW - Assembly week number LL - Wafer lot code
<CC>	Container code

Table 216: Abbreviations

12.4 Code ranges and values

Defined here are nRF5340 code ranges and values.

<PP>	Package	Size (mm)	Pin/Ball count	Pitch (mm)
QK	AQFN	7 x 7	94	0.4

Table 217: Package variant codes

<VV>	Flash (kB)	RAM (kB)
AA	1024	512

Table 218: Function variant codes

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

Table 219: Hardware version codes

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

Table 220: Production configuration codes

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

Table 221: Production version codes

<YY>	Description
[16 . . 99]	Production year: 2016 to 2099

Table 222: Year codes

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

Table 223: Week codes

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

Table 224: Lot codes

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

Table 225: Container codes

12.5 Product options

Defined here are the nRF5340 product options.

The following table lists the ordering code, as well as the minimum ordering quantity (MOQ).

Order code	MOQ
nRF5340-QKAA-R7	800
nRF5340-QKAA-R	3000
nRF5340-CLAA-R7	1500
nRF5340-CLAA-R	7000

Table 226: nRF5340 order codes

Order code	Description
nRF5340-DK	nRF5340 Development Kit

Table 227: Development tools order code

13 Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

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.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor ASA's products are not designed for use in life-critical medical equipment, support appliances, devices, or systems where malfunction of Nordic Semiconductor ASA's products can reasonably be expected to result in personal injury. Customer may not use any Nordic Semiconductor ASA's products in life-critical medical equipment unless adequate design and operating safeguards by customer's authorized officers have been made. Customer represents that, with respect to its applications, it has all the necessary expertise to create and implement safeguards that anticipate dangerous consequences of failures, monitor failures and their consequences, and lessen the likelihood of failures that might cause harm, and to take appropriate remedial actions. Customer agrees that prior to using or distributing any life-critical medical equipment that include Nordic Semiconductor ASA's products, customer will thoroughly test such systems and the functionality of such products as used in such systems.

Nordic Semiconductor ASA assumes no liability for applications assistance or the design of customers' products. Customers are solely responsible for the design, validation, and testing of its applications as well as for compliance with all legal, regulatory, and safety-related requirements concerning its applications.

Customer will fully indemnify Nordic Semiconductor ASA and its representatives against any damages, costs, losses, and/or liabilities arising out of customer's non-compliance with this section.

RoHS and REACH statement

Refer to www.nordicsemi.com for complete hazardous substance reports, material composition reports, and latest version of Nordic's RoHS and REACH statements.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

© Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

COMPANY WITH
QUALITY SYSTEM
CERTIFIED BY DNV GL
= ISO 9001 =