PRELIMINARY DATASHEET v0.6

nRF54LS05A and nRF54LS05B Wireless SoCs

nRF54LS05A and nRF54LS05B are part of the nRF54L Series. All wireless System-on-Chip (SoC) options in the series integrate an ultra-low power multiprotocol 2.4 GHz radio with MCU (Microcontroller Unit) functionality featuring a 128 MHz Arm® Cortex®-M33 processor. nRF54LS05A and nRF54LS05B support *Bluetooth*® LE and proprietary 2.4 GHz protocols.

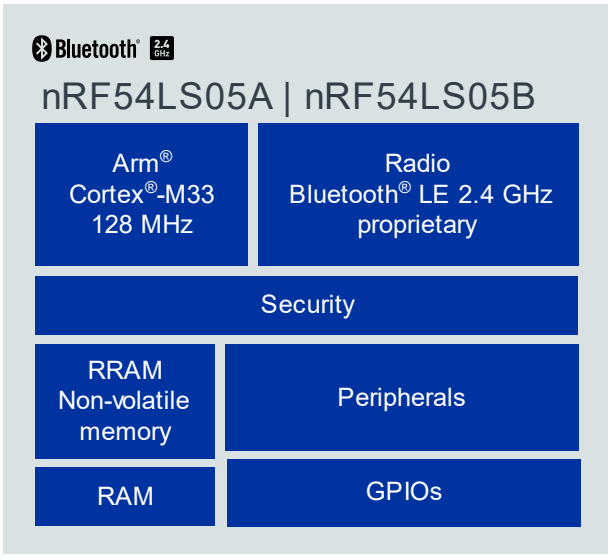
Nordic Semiconductor's proprietary technologies, such as low-leakage RAM and advanced multiprotocol radio design, enable ultra-low power consumption, reducing battery size or increasing lifetime.

Designed with versatility in mind, the nRF54L Series SoCs are suited to enable a broad range of applications. The devices integrate the peripherals expected in a wireless microcontroller, enabling a variety of products to be implemented with a single chip.

nRF54LS05A and nRF54LS05B are available in a range of pin-to-pin compatible QFN package options, enabling cost-optimized and flexible design across different application requirements.

Key features

- 128 MHz Arm® Cortex®-M33 processor
- 508 KB NVM and 64/96 KB RAM
- Multiprotocol 2.4 GHz radio supporting *Bluetooth*® LE, and 2.4 GHz proprietary modes (up to 4 Mbps)
- Three serial interfaces (SPI/TWI/UART)
- Standard set of interfaces, peripherals, and timers including Global RTC available in System OFF, ADC, PWM and QDEC
- Security: Baseline security configuration designed to support compliance with regulatory requirements for most *Bluetooth*® LE products
- QFN packages pin-to-pin compatible with selected SoCs



Power consumption highlights (Estimated)

Power mode	Current @ 3.0 V
Active with radio	
Bluetooth® LE TX 1 Mbps at 0 dBm	Estimated 4.8 mA
Bluetooth® LE TX 1 Mbps at +4 dBm	Estimated 6.6 mA
Bluetooth® LE RX 1 Mbps	Estimated 3.4 mA
Active with processing	
CPU CoreMark® from RRAM with cache	Estimated 2.0 mA
Sleep	
System ON IDLE with GRTC (XOSC) and 96 KB RAM	Estimated 1.2 µA
System OFF with GRTC wakeup	Estimated 0.7 µA
System OFF	Estimated 0.6 µA

Product variants

Part number	NVM	RAM
nRF54LS05A	508 KB	64 KB
nRF54LS05B	508 KB	96 KB

Feature overview

Radio

- *Bluetooth*® LE
  - LE 2M, LE 1M
- 2.4 GHz proprietary GFSK
  - 4 Mbps, 2 Mbps, 1 Mbps
- Single-ended antenna output (on-chip balun)
- 128-bit AES/ECB/CCM/AAR coprocessor
- Configurable TX power with 1 dBm step size from -10 dBm to maximum

Processor

- 128 MHz Arm® Cortex®-M33 CPU
  - MPU
  - Debug – SWD, DWT

Memory

- 508 KB NVM (RRAM)
- 64/96 KB RAM

Peripherals

- Three fully featured serial interfaces with EasyDMA, supporting I^2C, SPI controller/ peripheral, and UART
  - SPI up to 8 MHz (SPIM, SPIS)
  - TWI up to 400 kHz and I^2C compatible (TWIM, TWIS)
  - UART up to 1 Mbps (UARTE)
- SAADC with four channels
  - 14-bit at 31.25 ksps (oversampled)
  - 12-bit at 125 ksps (oversampled)
  - 10-bit at up to 1 Msps
- Global RTC can run in System OFF mode and implement a shared system timer (GRTC)
- One four-channel pulse width modulator (PWM) with autonomous waveform generation
- One quadrature decoder (QDEC)
- One watchdog timer (WDT)
- Three 32-bit timers with counter mode (TIMER)
- Temperature sensor (TEMP)

GPIO

- Up to 37 GPIO pins
- 8 MHz ports

Power supply and clock

- Single-inductor DC/DC converter
- Single 32 MHz crystal operation
- Optional 32.768 kHz crystal

Security

- Secure boot
- Physical protection – Tamper detectors (TAMPC)
- Hardware accelerated Bluetooth LE encryption

Key specifications

- 4 dBm maximum TX power
- -96 dBm RX sensitivity for 1 Mbps Bluetooth LE
- 500 CoreMark®, 3.90 CoreMark/MHz EEMBC CoreMark® executing from non-volatile memory
- 1.7 - 3.6 V supply and GPIO voltage
- -40°C to +85°C operating temperature

Package options

- QFN48 (QFAA) with 37 GPIO pins
  - 6x6 mm with 0.4 mm pitch

1 Revision history

Date	Version	Description
March 2026	0.6	Preliminary 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
Preliminary Datasheet	Applies to document versions up to 1.0. This document contains target specifications for product development.
Datasheet	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

The chapters describing peripherals include the following information:

- A description of the peripheral.
- The electrical specification tables, containing performance data which applies for the operating conditions described in Recommended operating conditions on page 563.

2.2.1 Peripheral naming conventions

Every peripheral has a unique capitalized name or an abbreviation of its name, such as TIMER, that is 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.

When there is more than one instance of a peripheral in a power domain, a two digit number Dn is added as a suffix to the peripheral name when constructing the peripheral instance name. For example, a peripheral named PERI with instance name "PERI`Dn`" is located in power domain `D`, and is instance number `n` in that domain. For a list of power domains, see Power domains on page 10.

The following are additional examples of peripheral instance names:

- PPIB00 is in the MCU domain (0), and is the first PPIB instance in the MCU domain (0).
- SPIS21 is in the PERI domain (2), and is the second SPIS instance of the PERI domain (1).

The peripheral instance name is also used in the CMSIS to identify the peripheral instance.

The domain digits x are listed in the following table.

Domain digit	Power domain
0	MCU
1	RADIO
2	PERI
3	LP

Table 2: Domain digit overview

2.3 Register overview table

The register overview table shows a summary of all peripheral registers.

The following table explains the columns of the register overview table.

Field	Description
Register	Name of register
Offset	Offset address from peripheral base address
Description	Short summary of intended use

Table 3: Register overview table

2.4 Register tables

Individual registers are described using register tables. These tables are composed 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.

2.4.1 Fields and values

The 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. These fields 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, such as 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, for example [0..4], indicating all values from 0 to 4.
- Implicit values. If no values are indicated in the Value column, all bit combinations are supported, or the field's translation and limitations are described in the text instead.

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

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

2.4.2 Permissions

Each register field can 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
R	Read only	Field can only be read. A write will be ignored.
W	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.
W0C	Write 0 to clear	Field can be read multiple times. A zero clears (set to zero) the corresponding bit in the register. Bits set to one are ignored.
W1C	Write 1 to clear	Field can be read multiple times. A one clears (set to zero) the corresponding bit in the register. Bits set to zero are ignored.
W1S	Write 1 to set	Field can be read multiple times. A one sets the corresponding bit in the register. Bits set to zero are ignored.
RME	Read Modify External	When read, a side effect occurs.

Table 4: Register field permission schemes

2.5 Registers

Register overview

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

2.5.1 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				J						I	H	G	F	E	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	FIELD0																																	
			Disabled	0																															
			NormalMode	1																															
			ExtendedMode	2																															
B	RW	FIELD1																																	
				This field is deprecated.																															
			Disabled	0																															
			Enabled	1																															
C	RW	FIELD2																																	

Bit number				31	30	29	28	27	26	25	24	23	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	E	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																															
			ValidRange	[2..7]																															
D	RW	FIELD3																																	
E	R	FIELD4																																	
F	W	FIELD5																																	
G	RW W1C	FIELD6																																	
H	RW W0C	FIELD7																																	
I	RW RME	FIELD8																																	
J	RW	PRIVATE																																	

3 Product overview

nRF54LS05A and nRF54LS05B are pin compatible ultra-low power System on Chip (SoC), with different memory configurations.

The multiprotocol 2.4 GHz transceiver supports Bluetooth Low Energy and allows for the implementation of proprietary 2.4 GHz protocols.

The main processing unit is an Arm Cortex-M33 processor running at up to 128 MHz, supported by non-volatile RRAM and RAM memory.

The Arm Cortex-M33 has a memory protection unit (MPU) for application security.

The peripheral set offers a variety of analog and digital functionality enabling single-chip implementation of a wide range of applications.

The device non-volatile memory has a boot region that can be made immutable before the CPU starts up. Boot initiated from an immutable source allows subsequent boot steps to be performed by authenticated code.

The debug access port can be enabled or disabled to allow debugging. The non-volatile memory can be protected against erasing, providing protection from unauthenticated repurposing. Authenticated debug access control, such as facilitating the Arm ADAC architecture, is supported through a hardware mailbox. The mailbox allows on-chip firmware to authenticate the debug host before enabling the device debug interface.

3.1 Block diagram

The block diagram illustrates the overall system.

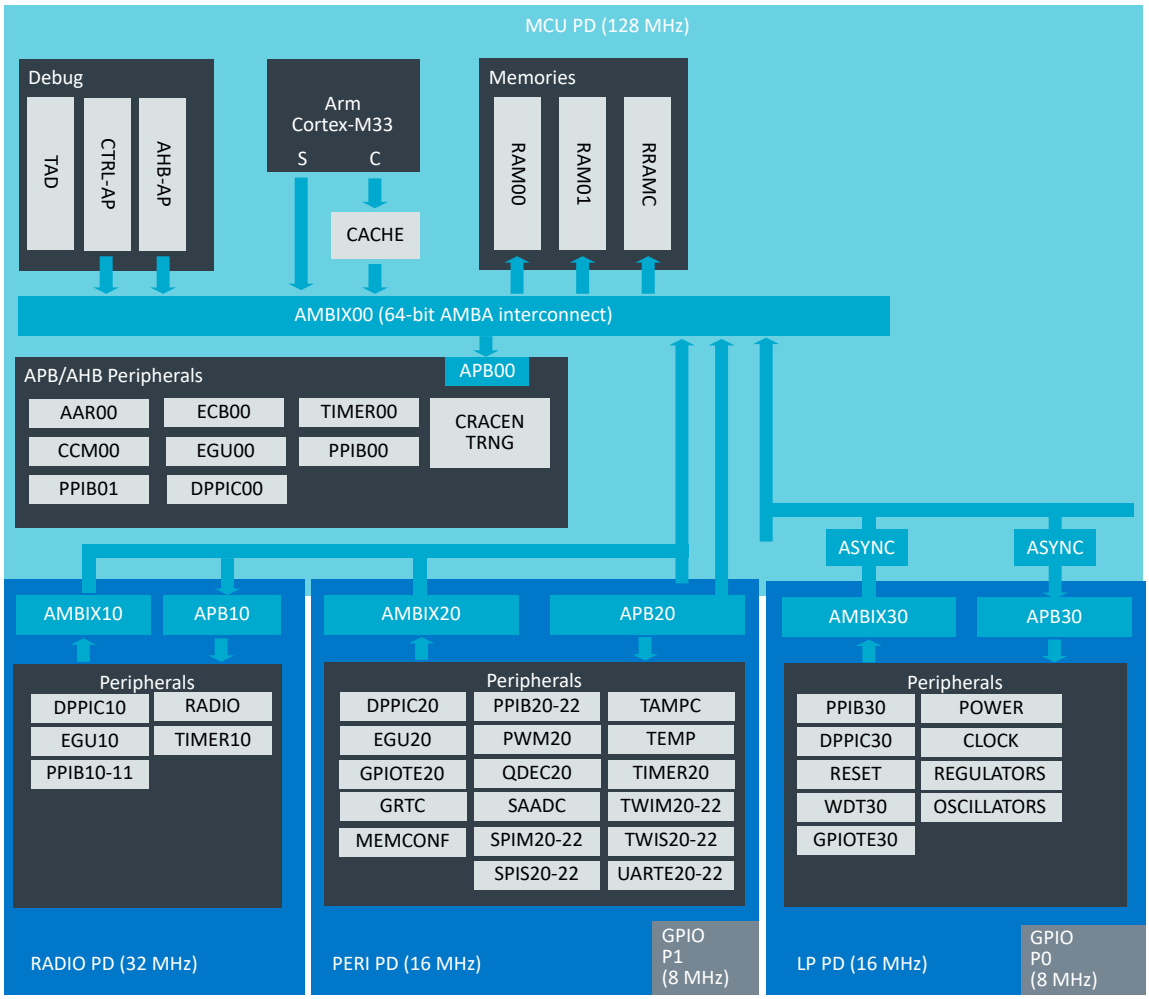


Figure 1: Block diagram

3.2 Memory and package overview

Memory	Device	
	nRF54LS05A	nRF54LS05B
Non-volatile memory (RRAM)	508 KB	508 KB
Random access memory (RAM)	64 KB	96 KB

Table 5: Device memory options

Feature		Package
		QFN48
Pins	GPIO pins	37
	Wakeup-pins	37
	Analog input pins	4

Table 6: Package variants

3.3 Power domains

Multiple power domains ensure low-power operation.

The MCU domain contains an Arm Cortex-M33. The CPU is connected to a debug system that allows debug. The CPU executes program code from RRAM through an instruction cache. Data is stored in single-cycle RAM that is divided into multiple bus subordinates forming a continuous RAM space in the memory map. High-speed peripherals are also found in the MCU domain.

There are three additional domains that have peripherals allocated to them. They are the following:

- Radio domain – Contains the short-range radio and supporting peripherals used by the radio protocol stack. It runs at 32 MHz synchronously with the MCU domain.
- Peripheral domain – Contains most peripherals. It runs at 16 MHz synchronously with the MCU domain.
- Low-power domain – Contains peripherals for ultra-low power modes and can be used to wake the rest of the system, even when the peripheral domain is powered off. It runs at 16 MHz asynchronously to the MCU domain.

Each domain is mapped to one APB bus and can be powered independently. EasyDMA traffic from each domain is aggregated in a local AMBIX interconnect and can access RAM in the MCU domain.

Power domains have their own GPIO ports. GPIO pins can be used by peripherals in the same power domain. For exceptions, see GPIO — General purpose input/output on page 182 and pin assignments.

3.4 Address format

Addresses in the system memory map follow the address format described in the following tables.

Address bits	Description	Enumeration
[28:0]	Address space	
[31:29]	Address regions	0: Program memory 1: Data memory 2: Peripherals/APB space 7: CPU internal peripherals, like Arm Cortex private peripheral bus (PPB)

Table 7: Address regions format

The program and data memory address format is described in the following table.

Address bits	Description	Enumeration
[23:0]	Address space	
[28:24]	Reserved	Set to zero
[31:29]	Address regions	0: Program memory 1: Data memory

Table 8: Program memory and data memory address format

The peripheral address format is described in the following table.

Address bits	Description	Enumeration
[11:0]	Peripheral address space	
[17:12]	Peripheral subordinate index	Used for configuring the , as the index *n* of register SPU.PERIPH[*n*].PERM.
[20:12]	Interrupt vector number	The index in the interrupt vector table
[23:18]	Peripheral APB bus number	1: APB peripherals in MCU power domain 2: APB peripherals in RADIO power domain 3: APB peripherals in PERI power domain 4: APB peripherals in LP power domain
[27:24]	Reserved	Set to zero
[28]	Reserved	Set to zero
[31:29]	Address regions	2: Peripherals on APB bus

Table 9: Peripheral address format

3.5 Memory

The CPU and peripherals with EasyDMA can access memory through the AMBIX interconnects. The same interconnect is used by the CPU to access peripheral registers. The following figure is a simplified interconnect diagram.

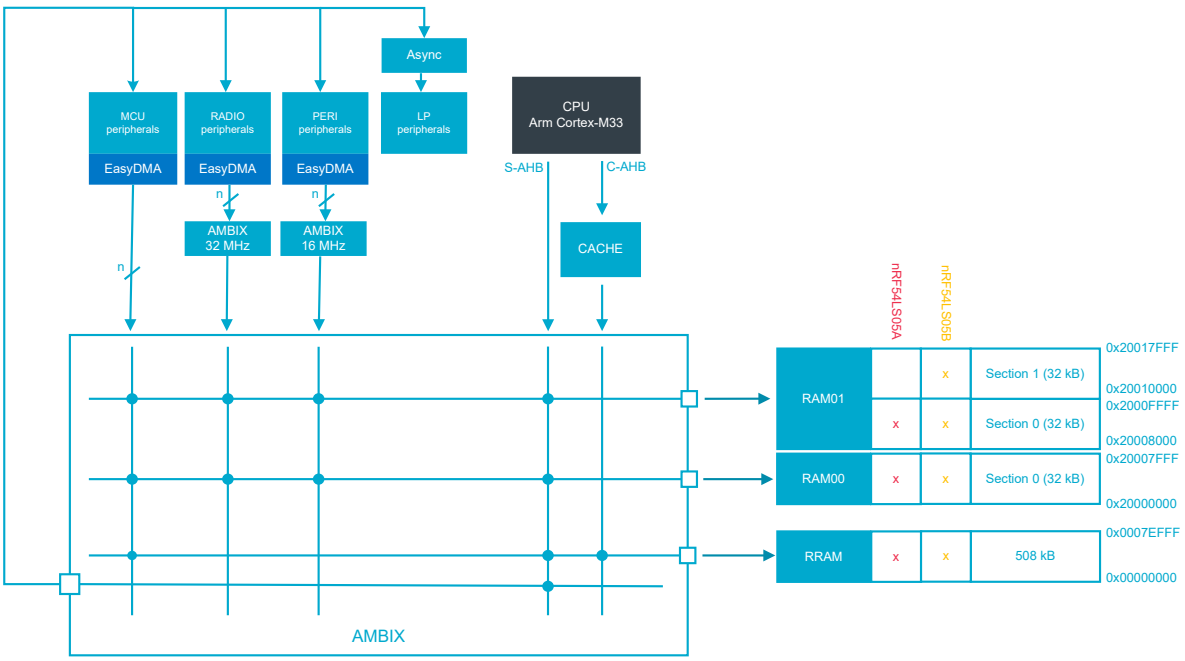


Figure 2: Memory layout

See Block diagram on page 8, AMBA interconnect (AMBIX) on page 19, and EasyDMA on page 19 for more information about the AMBIX interconnects and EasyDMA.

3.5.1 RAM — Random access memory

The device RAM has regions arranged in one contiguous memory range, accessible from both the CPU and peripherals.

Each RAM region has separate power control for System ON and System OFF mode. This preserves RAM contents in sleep modes or powers off RAM to save power. The sections are illustrated in Memory layout on page 12, and the register interface is described in MEMCONF — Memory configuration on page 30.

3.5.2 NVM — Non-volatile memory

The CPU can read from non-volatile memory (RRAM) an unlimited number of times, but is restricted in how it writes to memory and the number of writes it can perform.

Writing to RRAM is managed by the RRAM controller (RRAMC), see RRAMC — Resistive random access memory controller on page 32.

The Arm Cortex-M33 CPU can access RRAM using the C-AHB (code) bus as shown in Memory layout on page 12. The code bus (C-AHB) interface is used for any instruction fetch or data access fetch to the code region of the Arm memory model. C-AHB bus access is cached, see CACHE — Instruction/data cache on page 21.

3.5.3 Memory map

The complete memory map is shown in the following figure.

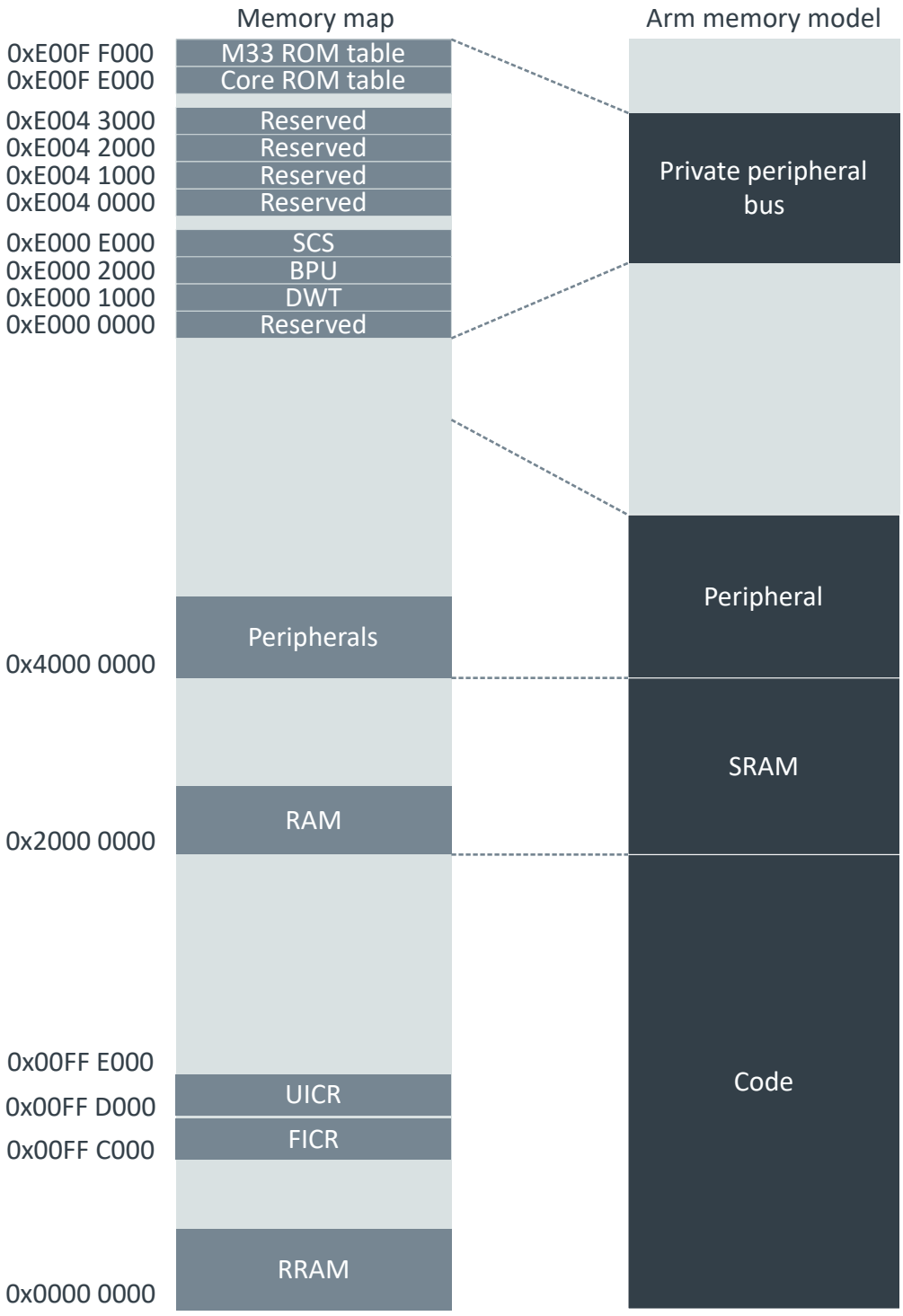


Figure 3: Memory map

3.5.4 Instantiation

ID	Base address	Instance	Description
66	0x40042000	DPPIC00	DPPI controller DPPIC00
68	0x40044000	PPIB00	PPI bridge PPIB00
69	0x40045000	PPIB01	PPI bridge PPIB01
74	0x4004A000	AAR00	Accelerated address resolver 00
74	0x4004A000	CCM00	AES CCM mode encryption CCM00, running of HCLKCORE

ID	Base address	Instance	Description
75	0x4004B000	ECB00	AES ECB mode encryption 00
78	0x4004E000	RRAMC	RRAM Non-Volatile Memory Controller
82	0x40052000	CTRLAP	Control access port CPU side
83	0x40053000	TAD	Trace and debug control
85	0x40055000	TIMER00	Timer TIMER00
88	0x40058000	EGU00	Event generator unit EGU00
89	0x40059000	CRACEN	True Random Number Generator
89	0x40059000	CRACENCORE	True Random Number Generator
130	0x40082000	DPPIC10	DPPI controller DPPIC10
131	0x40083000	PPIB10	PPI bridge PPIB10
132	0x40084000	PPIB11	PPI bridge PPIB11
133	0x40085000	TIMER10	Timer TIMER10
135	0x40087000	EGU10	Event generator unit EGU10
138	0x4008A000	RADIO	2.4 GHz radio RADIO
194	0x400C2000	DPPIC20	DPPI controller DPPIC20
195	0x400C3000	PPIB20	PPI bridge PPIB20
196	0x400C4000	PPIB21	PPI bridge PPIB21
197	0x400C5000	PPIB22	PPI bridge PPIB22
198	0x400C6000	SPIM20	SPI controller SPIM20
198	0x400C6000	SPIS20	SPI peripheral SPIS20
198	0x400C6000	TWIM20	Two-wire interface controller TWIM20
198	0x400C6000	TWIS20	Two-wire interface target TWIS20
198	0x400C6000	UARTE20	Universal asynchronous receiver/transmitter UARTE20
199	0x400C7000	SPIM21	SPI controller SPIM21
199	0x400C7000	SPIS21	SPI peripheral SPIS21
199	0x400C7000	TWIM21	Two-wire interface controller TWIM21
199	0x400C7000	TWIS21	Two-wire interface target TWIS21
199	0x400C7000	UARTE21	Universal asynchronous receiver/transmitter UARTE21
200	0x400C8000	SPIM22	SPI controller SPIM22
200	0x400C8000	SPIS22	SPI peripheral SPIS22
200	0x400C8000	TWIM22	Two-wire interface controller TWIM22
200	0x400C8000	TWIS22	Two-wire interface target TWIS22
200	0x400C8000	UARTE22	Universal asynchronous receiver/transmitter UARTE22
201	0x400C9000	EGU20	Event generator unit EGU20
202	0x400CA000	TIMER20	Timer TIMER20
207	0x400CF000	MEMCONF	Memory Configuration MEMCONF
210	0x400D2000	PWM20	Pulse width modulation unit PWM20
213	0x400D5000	SAADC	Successive approximation analog-to-digital converter SAADC
215	0x400D7000	TEMP	Temperature sensor TEMP
216	0x400D8200	P1	General purpose input and output, port P1
218	0x400DA000	GPIOTE20	8 channels and 1 interrupts for GPIO port P1 GPIO tasks and events GPIOTE20
224	0x400E0000	QDEC20	Quadrature decoder QDEC20
226	0x400E2000	GRTC	Global RTC GRTC
239	0x400EF000	TAMPC	Tamper controller TAMPC
258	0x40102000	DPPIC30	DPPI controller DPPIC30
259	0x40103000	PPIB30	PPI bridge PPIB30
264	0x40108000	WDT30	Watchdog timer WDT30
266	0x4010A000	P0	General purpose input and output, port P0
268	0x4010C000	GPIOTE30	4 channels and 1 interrupts for GPIO port P0 GPIO tasks and events GPIOTE30
270	0x4010E000	CLOCK	Clock control

ID	Base address	Instance	Description
270	0x4010E000	POWER	Power control
270	0x4010E000	RESET	Reset status
288	0x40120000	OSCILLATORS	Oscillator control
288	0x40120000	REGULATORS	Regulator control
N/A	0x00FFC000	FICR	Factory information configuration
N/A	0x00FFD000	UICR	User information configuration

Table 10: Instantiation table

4 Application core

4.1 Arm Cortex-M33 CPU

4.1.1 CPU

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

This processor has the following features that enable energy-efficient arithmetic and high-performance signal processing:

- Hardware divide
- Memory Protection Unit (MPU)
- Stack limit checking

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

Instruction cache on the C-bus (code bus) of the Cortex-M33 CPU improves performance when fetching instructions (or data) from internal non-volatile memory. For more information on cache, see CACHE — Instruction/data cache on page 21. CPU performance parameters including wait states for configurations, CPU current consumption and efficiency, and processing power and efficiency based on the CoreMark benchmark can be found in CPU Electrical specification on page 554.

4.1.1.1 CPU and support module configuration

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

Option	Description	Implemented
WIC	Wakeup Interrupt Controller	No
Endianness	Memory system endianness	Little endian
DWT	Data Watchpoint and Trace	Yes

Table 11: Core options

Module	Description	Implemented
MPU	Number of MPU regions	4
SAU	Number of SAU regions	0
FPU	Floating-point unit	No
DSP	Digital Signal Processing Extension	No
Arm TrustZone® for Armv8-M	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	No
BPU	Breakpoint Unit	Yes
INITVTOR	Initial vector table offset	0x00000000

Table 12: Modules

4.1.2 CPUC — CPU control

CPUC controls elements of the Arm Cortex-M33 processor such as lock certain features of the CPU and prevent them from being modified.

CPUC holds a CPU identifier CPUID, used in the system to uniquely identify the processing unit of a core.

In addition, CPUC holds a LOCK register, which is used to lock certain CPU features and prevent them from being modified. One example is the LOCK.LOCKSAU field. When set to `Locked`, this prevents further modifications to the SAU registers.

4.1.2.1 Registers

Instances

Instance	Domain	Base address	Description
CPUC	APPLICATION	0xE0080000	Cortex-M33 configuration

Register overview

Register	Offset	Description
LOCK	0x500	Register to lock the certain parts of the CPU from being modified.
CPUID	0x504	The identifier for the CPU in this subsystem.

4.1.2.1.1 LOCK

Address offset: 0x500

Register to lock the certain parts of the CPU from being modified.

Each bit can only be written once and can only be changed from 0 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				E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	LOCKVTORAIRCRS			Locks both the Vector table Offset Register (VTOR) and Application Interrupt and Reset Control Register (AIRCR) for secure mode.
			NotLocked	0	Both VTOR and AIRCR can be changed.
			Locked	1	Prevents changes to both VTOR and AIRCR.
B	RW	LOCKVTORNS			Locks the Vector table Offset Register (VTOR) for non-secure mode.
			NotLocked	0	VTOR can be changed.
			Locked	1	Prevents changes to VTOR.
C	RW	LOCKMPUS			Locks the Memory Protection Unit (MPU) for secure mode.
			NotLocked	0	MPU registers can be changed.
			Locked	1	Prevents changes to MPU registers.
D	RW	LOCKMPUNS			Locks the Memory Protection Unit (MPU) for non secure mode.
			NotLocked	0	MPU registers can be changed.
			Locked	1	Prevents changes to MPU registers.
E	RW	LOCKSAU			Locks the Security Attribution Unit (SAU)
			NotLocked	0	SAU registers can be changed.
			Locked	1	Prevents changes to SAU registers.

4.1.2.1.2 CPUID

Address offset: 0x504

The identifier for the CPU in 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 A 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	CPUID			The CPU identifier.

4.1.3 Arm Cortex-M33 Peripherals

4.1.3.1 Instantiation

ID	Base address	Instance	Description
28	0x4001C000	SWI00	Software interrupt SWI00
29	0x4001D000	SWI01	Software interrupt SWI01
30	0x4001E000	SWI02	Software interrupt SWI02
31	0x4001F000	SWI03	Software interrupt SWI03
N/A	0xE0080000	CPUC	Cortex-M33 configuration
N/A	0xE0082000	ICACHE	Instruction cache

Table 13: Instantiation table

4.2 Core components

4.2.1 AMBA interconnect (AMBIX)

The AMBA interconnect (AMBIX) is a multilayer-capable bus interconnect that provides low latency access from Managers to Subordinates.

Manager and Subordinate connections are arranged in Manager and Subordinate pairs, allowing for a sparse bus matrix. The interconnect supports multiple concurrent transactions when targeting different Subordinates.

Some peripherals do not have the opportunity to pause incoming data. Being a low priority bus manager might cause loss of data for such peripherals upon bus contention. To avoid bus contention when using multiple bus managers, follow these guidelines:

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

4.2.2 EasyDMA

EasyDMA is a module implemented by some peripherals as a bus manager for direct access to RAM. Not all EasyDMA instances can access non-volatile memory.

A peripheral can implement multiple EasyDMA instances to provide dedicated channels. For example, a channel can be dedicated for reading and writing data between the peripheral and RAM. This concept is illustrated in the following example figure, where READER is reading data from RAMc, while WRITER is writing data to RAMa and RAMb (each RAM being separate bus subordinates).

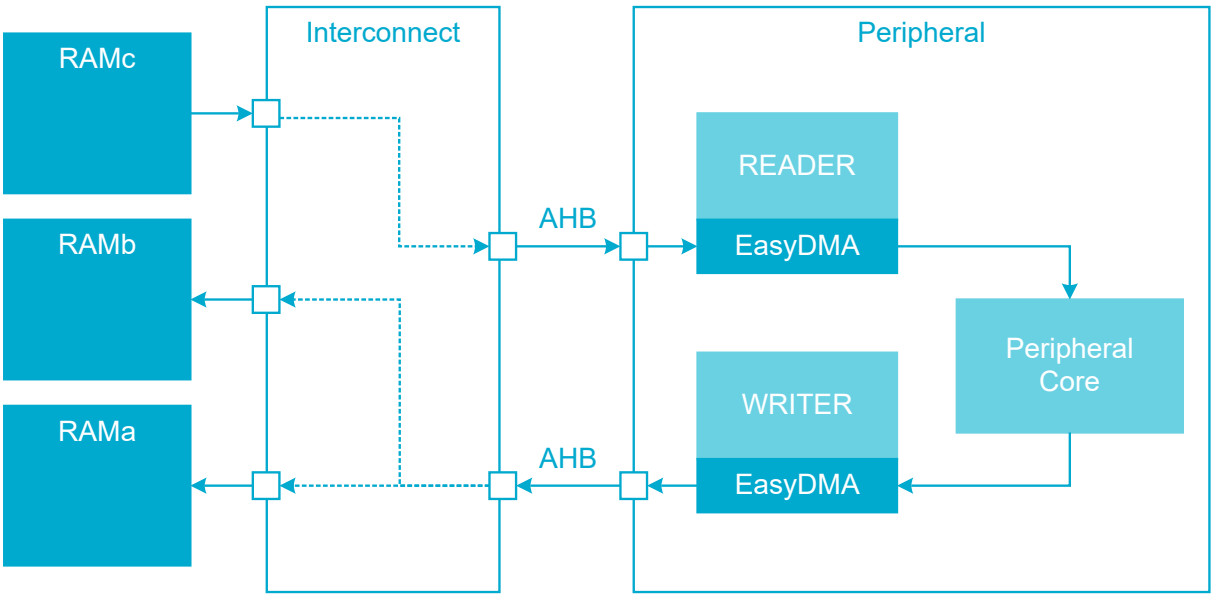


Figure 4: EasyDMA example

4.2.2.1 EasyDMA channel implementation

A typical EasyDMA channel is implemented in the following way.

```
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 channel is for reading called READER, and one for writing called WRITER. When the peripheral starts, it performs the following tasks.

1. Reads 5 B from the readerBuffer located in RAM at address 0x20000000.
2. Processes the data.
3. Writes up to 6 B back to the writerBuffer located in RAM at address 0x20000005.

The memory layout of these buffers is illustrated is shown in the following figure.

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

Figure 5: EasyDMA memory layout

The specified size of the WRITER.MAXCNT register must not be larger than the actual size of the buffer (writerBuffer). This prevents the channel from overflowing the writerBuffer.

Once an EasyDMA transfer is complete, the CPU reads the AMOUNT register to see how many bytes were transferred. For example, the CPU can read the MYPERIPHERAL.WRITER.AMOUNT register to see how many bytes WRITER wrote to RAM.

Note: A READER or WRITER PTR register must point to a valid memory region before using EasyDMA. The reset value of a PTR register is not guaranteed to point to valid memory. See Memory on page 11 for more information about the memory regions and EasyDMA connectivity.

4.2.2.2 EasyDMA error handling

Errors can occur during DMA handling.

If READER.PTR or WRITER.PTR is not pointing to a valid memory region, an EasyDMA transfer could HardFault or cause RAM corruption. See Memory on page 11 for more information about the different memory regions.

An EasyDMA channel is an AHB bus Manager. If several AHB Managers try to access the same AHB Subordinate at the same time, AHB bus congestion can occur. Depending on the peripheral, the peripheral could either stall and wait for access to be granted, or lose data.

4.2.2.3 Array list

EasyDMA can operate in Array List mode.

The Array List mode is implemented in channels where the LIST register is available.

The array list is not able to 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 is implemented with the data structure ArrayList_type. This is illustrated in the following code example 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 includes a buffer that is equal in size to the READER.MAXCNT register. EasyDMA uses the READER.MAXCNT register to determine when the buffer is full.

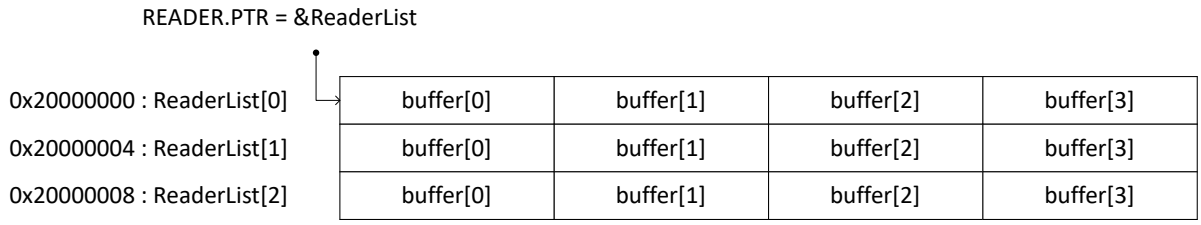


Figure 6: EasyDMA array list

4.2.3 CACHE — Instruction/data cache

The cache is a direct mapped cache. Both instruction and data accesses towards NVM memory are cached.

The cache has the following features:

- 4x64-bit cache line
- Ability to enable/disable cache at run-time
- Writes to cached memory are write-around and invalidate the cache line
- Manual invalidation
- Performance hit/miss counter registers for profiling CACHE operations

The cache must be enabled by the ENABLE register.

4.2.3.1 Architecture

The following figure shows the cache architecture.

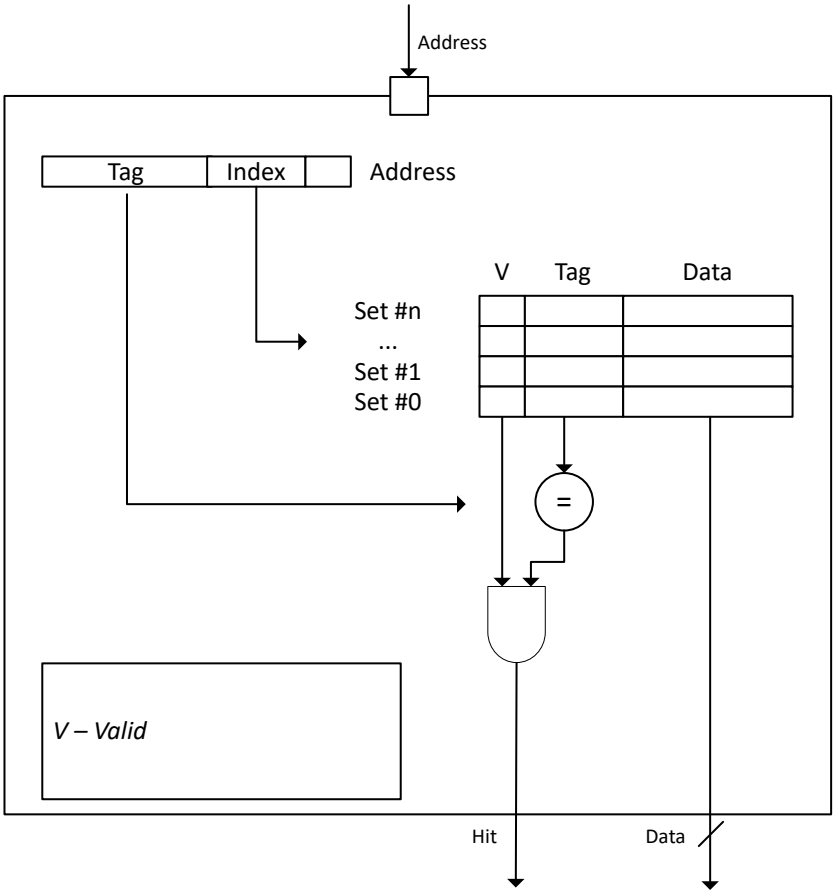


Figure 7: Cache overview

Bit	Name	Description
V	Valid	Indicates if a cache entry is valid. All V fields are cleared when enabling the cache, invalidating the cache, or when changing CACHE mode.

4.2.3.2 Profiling

The cache provides a scoreboard that tracks the hits and misses within the cache.

The results are available through a set of registers that can be used to indicate how well the cache is performing.

Profiling is enabled using PROFILING.ENABLE. All profiling counters can be cleared at any time using PROFILING.CLEAR. After being cleared, the counters will increment, according to the rules in the table below, at the next instruction- or data fetch.

Profiling counter	Description
HIT	Incremented on a cache hit
MISS	Incremented on a cache miss (not counting write misses)

Table 14: Profiling counters

4.2.3.3 Registers

Instances

Instance	Domain	Base address	Description
ICACHE	APPLICATION	0xE0082000	Instruction cache

Configuration

Instance	Domain	Configuration
ICACHE	APPLICATION	Cache size: 8 KB. Sets: 128. Data unit: 64 bits. Line width = 4 data units. Data bus width : 0..63

Register overview

Register	Offset	Description
TASKS_INVALIDATECACHE	0x008	Invalidate the cache.
STATUS	0x400	Status of the cache activities.
ENABLE	0x404	Enable cache.
PROFILING.ENABLE	0x414	Enable the profiling counters.
PROFILING.CLEAR	0x418	Clear the profiling counters.
PROFILING.HIT	0x41C	The cache hit counter for cache region.
PROFILING.MISS	0x420	The cache miss counter for cache region.

4.2.3.3.1 TASKS_INVALIDATECACHE

Address offset: 0x008

Invalidate the cache.

The STATUS is updated for this task. This task can be triggered only when the STATUS 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	W	TASKS_INVALIDATECACHE			Invalidate the cache. The STATUS is updated for this task. This task can be triggered only when the STATUS is ready.
			Trigger	1	Trigger task

4.2.3.3.2 STATUS

Address offset: 0x400

Status of the cache activities.

Indicates status of the activities initiated using respective 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	R	READY			Ready status.
			Ready	0	Activity is done and ready for the next activity.
			Busy	1	Activity is in progress.

4.2.3.3.3 ENABLE

Address offset: 0x404

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

4.2.3.3.4 PROFILING.ENABLE

Address offset: 0x414

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

4.2.3.3.5 PROFILING.CLEAR

Address offset: 0x418

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

4.2.3.3.6 PROFILING.HIT

Address offset: 0x41C

The cache hit counter for cache region.

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

4.2.3.3.7 PROFILING.MISS

Address offset: 0x420

The cache miss counter for cache region.

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

4.2.4 FICR — Factory information configuration registers

Factory information configuration registers (FICR) are pre-programmed in factory. These registers contain chip-specific information and configuration.

4.2.4.1 Registers

Instances

Instance	Domain	Base address	Description
FICR	GLOBAL	0x00FFC000	Factory information configuration

Register overview

Register	Offset	Description
INFO.CONFIGID	0x300	Configuration identifier
INFO.DEVICEID[n]	0x304	Device identifier
INFO.UUID[n]	0x30C	128-bit Universally Unique IDentifier (UUID).
INFO.PART	0x31C	Part code
INFO.VARIANT	0x320	Function variant code, Hardware version code, and Production configuration code encoded as ASCII.
INFO.PACKAGE	0x324	Package option
INFO.RAM	0x328	RAM size (KB)
INFO.RRAM	0x32C	RRAM size (KB)
ER[n]	0x380	Common encryption root key, word n
IR[n]	0x390	Common identity root key, word n
DEVICEADDRTYPE	0x3A0	Device address type
DEVICEADDR[n]	0x3A4	Device address n
TRIMCNF[n].ADDR	0x400	Address of the register which will be written
TRIMCNF[n].DATA	0x404	Data to be written into the register
XOSC32MTRIM	0x620	XOSC32M capacitor selection trim values
XOSC32KTRIM	0x624	XOSC32K capacitor selection trim values

4.2.4.1.1 INFO

Device info

4.2.4.1.1.1 INFO.CONFIGID

Address offset: 0x300

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

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

Address offset: 0x304 + (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.

4.2.4.1.1.3 INFO.UUID[n] (n=0..3)

Address offset: 0x30C + (n × 0x4)

128-bit Universally Unique IDentifier (UUID).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A 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	UUID			Device UUID [n]. The DEVICE.UUID[0] contains the least significant bits of the device identifier.

4.2.4.1.1.4 INFO.PART

Address offset: 0x31C

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 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	PART			Part code
			N54LS05A	0x054BE10A	N54LS05A
			N54LS05B	0x054BE10B	N54LS05B
			Unspecified	0xFFFFFFFF	Unspecified

4.2.4.1.1.5 INFO.VARIANT

Address offset: 0x320

Function variant code, Hardware version code, and Production configuration code encoded as ASCII.

The VARIANT register is an ASCII encoded string of the form <VV> <H> <P>, where <VV> is the Function variant code, <H> is the Hardware version code, and <P> is the Production configuration code.

See the Ordering information chapter for more details on the function variant code, hardware version code, and production configuration 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 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	VARIANT																																	
			AAB0	0x41414230																															
			Unspecified	0xFFFFFFFF																															

4.2.4.1.1.6 INFO.PACKAGE

Address offset: 0x324

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																															
A	R	PACKAGE																																	
			Unspecified	0xFFFFFFFF																															

4.2.4.1.1.7 INFO.RAM

Address offset: 0x328

RAM size (KB)

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	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	RAM																																	
			K96	0x60																															
			K64	0x40																															
			Unspecified	0xFFFFFFFF																															

4.2.4.1.1.8 INFO.RRAM

Address offset: 0x32C

RRAM size (KB)

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

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

Address offset: 0x380 + (n × 0x4)

Common encryption root key, word n

Used to generate keys as recommended by the Bluetooth Core Specification

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

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

Address offset: 0x390 + (n × 0x4)

Common identity root key, word n

Used to generate keys as recommended by the Bluetooth Core Specification

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

4.2.4.1.4 DEVICEADDRTYPE

Address offset: 0x3A0

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

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

Address offset: 0x3A4 + (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.

4.2.4.1.6 TRIMCNF[n].ADDR (n=0..63)

Address offset: 0x400 + (n × 0x8)

Address of the register which will 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 A 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

4.2.4.1.7 TRIMCNF[n].DATA (n=0..63)

Address offset: 0x404 + (n × 0x8)

Data to be written into the 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 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	Data			Data

4.2.4.1.8 XOSC32MTRIM

Address offset: 0x620

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 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	R	SLOPE			Slope trim factor on twos complement form -256 = '1_0000_0000' and +255 = '0_1111_1111'
B	R	OFFSET			Offset trim factor on integer form

4.2.4.1.9 XOSC32KTRIM

Address offset: 0x624

XOSC32K capacitor selection trim values

Note: To enable the optional internal capacitors on XL1 and XL2 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	B	B	B	B	B								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	SLOPE																									Slope trim factor on twos complement form								
B	R	OFFSET																									Offset trim factor on integer form								

4.2.5 MEMCONF — Memory configuration

MEMCONF provides power control for RAM blocks.

Each RAM block can independently power up or power down in System ON and System OFF mode. RAM blocks can contain multiple RAM sections. For more information about System ON and System OFF modes, see Power and clock management on page 51. For an overview of available RAM blocks and RAM sections, see Memory on page 11.

MEMCONF registers are used for configuring the following:

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

In System OFF mode, a RAM section is retained by configuring the corresponding MEM[i] field of the RET register.

In System ON mode, retention and accessibility for a RAM section is configured in the corresponding MEM[i] field of register POWER[n].CONTROL (n=0..1) on page 31.

A complete list of blocks (RET.MEM[i], RET2.MEM[i], and CONTROL.MEM[i]) are found in the following table.

Block number (index i in MEMCONF.POWER)	RAM section	RET reset value	CONTROL reset value
0	RAM00 section 0	1	1
1	RAM01 section 0	1	1
2	RAM01 section 1	1	1
33	ICACHE tag + data 1:0	1	1

Table 15: Memory block overview with MEMCONF.POWER configuration

The following table summarizes the behavior of the CONTROL and RET fields when a power domain is powered on or off. The RAM section can be used to read and write data when it is powered. The RAM section is retained during System OFF.

Configuration			RAM section status	
Power mode	CONTROL	RET	Powered	Retained
System OFF	Any value	Off	No	No
System OFF	Off	On	No	No
System OFF	On	On	No	Yes
System ON IDLE	Off	Any value	No	No
System ON IDLE	On	Any value	No	Yes
System ON RUN	Any value	Any value	Yes	Yes

Table 16: RAM section configuration

The advantage of not retaining RAM content is reduced overall current consumption.

See chapter Memory on page 11 for more information on RAM sections.

Note: CACHE — Instruction/data cache on page 21 must be disabled when the ICACHE memory block is turned off, and only enabled after the ICACHE memory block is turned on.

4.2.5.1 Registers

Instances

Instance	Domain	Base address	Description
MEMCONF	GLOBAL	0x400CF000	Memory Configuration MEMCONF

Configuration

Instance	Domain	Configuration
MEMCONF	GLOBAL	

Register overview

Register	Offset	Description
POWER[n].CONTROL	0x500	Control memory block power.
POWER[n].RET	0x508	RAM retention for RAM [n].

4.2.5.1.1 POWER[n].CONTROL (n=0..1)

Address offset: 0x500 + (n × 0x10)

Control memory block power.

Where n = 0 for memory blocks 0 to 31 and n = 1 for memory blocks 32 to 63.

Bit number				31	30	29	28	27	26	25	24	23	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																															
A-f	RW	MEM[i] (i=0..31)																																	
			Off	0																															
			On	1																															

Description
Keep the memory block MEM[i] on or off when in System ON mode.
RAM blocks powered off this way will not be retained. All RAM blocks will be off in System OFF mode.
Power down
Power up

4.2.5.1.2 POWER[n].RET (n=0..1)

Address offset: 0x508 + (n × 0x10)

RAM retention for RAM [n].

Where n = 0 for RAM blocks 0 to 31 and n = 1 for RAM blocks 32 to 63.

Bit number				31	30	29	28	27	26	25	24	23	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																															
A-f	RW	MEM[i] (i=0..31)																																	
			Off	0																															
			On	1																															

Description
Keep the RAM block MEM[i] retained when in System OFF mode.
All other RAM will be off in System OFF mode.
Retention off
Retention on

4.2.6 RRAMC — Resistive random access memory controller

The resistive random access memory controller (RRAMC) is used for writing the internal RRAM memory, , and the user information configuration registers (UICR).

The main features of RRAMC are:

- Write and overwrite without erasing
- 64-bit word line with built in error correction code (ECC), detecting and correcting up to two bit errors per line
- Automatic standby or power-down modes
- One-time programmable (OTP) protection for user information configuration registers (UICR)
- Optional immutable boot region protection

4.2.6.1 Reading from RRAM

RRAM can be read using any natural alignment.

Read and execute operations are cachable through CACHE — Instruction/data cache on page 21. For execution performance figures, see CPU on page 16.

Low latency mode

The low power register allows making trade-offs between latency and power consumption. By default, RRAMC goes into PowerDown mode, so the wakeup time is variable. To enable a sleep mode with faster wake-up, configure Standby mode using register POWER.LOWPOWERCONFIG.MODE. In combination with Constant Latency sub-power mode, this ensures the lowest latency.

4.2.6.2 Writing to RRAM

When writing is enabled in register CONFIG.WEN, and CONFIG.WRITEBUFSIZE is set to `Unbuffered`, RRAM is written using any natural alignment (byte, half-word, 32-bit, or 64-bit).

RRAMC always writes full wordlines. When data is written to RRAM, the entire wordline is written and ECC is updated, even if only a single bit is changed. This has an effect on the endurance of the RRAM, as each write operation counts as a write to all bits in the wordline.

RRAMC is able to write both `0` and `1` to any bit in RRAM, even if that bit has been written before.

When writing with CONFIG.WRITEBUFSIZE set to `Unbuffered`, the written data (byte, half-word, 32-bit, or 64-bit) is committed to RRAM immediately.

4.2.6.2.1 Buffered RRAM write

RRAMC enables fast buffered writes for contiguous memory regions.

RRAMC has an internal write-buffer that can be configured using CONFIG.WRITEBUFSIZE.

When buffered writes are enabled, RRAMC will collect as much data as possible in the internal write-buffer, before bulk-committing the buffer to RRAM.

When committing to RRAM, the commit operation updates only the data in RRAM memory that has modified values. Values that have not been altered remain unchanged in RRAM.

To use buffered writes, perform the following operations:

1. Enable writing using CONFIG.WEN, and configure CONFIG.WRITEBUFSIZE.
2. Write to RRAM memory in incrementing address order.

RRAMC commits the write-buffer when either of the following occurs:

- The write-buffer is full
- The address written is outside the buffer area
- There is a read operation from a 64-bit word line in the buffer that has already been written to

RRAMC stalls while the commit takes place, and additional wait-states can be observed for the bus access.

In addition to the automatic commit, a commit can also be triggered by the following:

- After a time-out waiting for a new write operation, configured in READYNEXTTIMEOUT
- When the COMMITWRITEBUF task is triggered

The manual triggers are useful in situations where it is crucial to ensure that the write buffer has been committed. Register BUFSTATUS.WRITEBUFEMPTY can be used to check if the write-buffer is empty, or if it contains uncommitted data.

> **Note:** The internal write-buffer is volatile, and data loss may occur during a power failure or when entering System OFF mode with uncommitted data in the buffer. Having uncommitted data in the internal write-buffer will keep the RRAM active, and thus increase power consumption during sleep mode.

4.2.6.3 Erasing RRAM

RRAMC provides a mechanism to erase the whole RRAM in one operation by using the ERASE.ERASEALL register.

When ERASEALL is triggered, RRAMC will write `0xFFFFFFFF` to the entire RRAM memory, including user information configuration registers (UICR) . ERASEALL will not erase the factory information configuration registers (FICR).

This functionality can be blocked by ERASE protection. For details, see CTRL-AP — Control access port on page 531.

Note: Unlike the CTRL-AP ERASEALL operation that can be activated from a debugger, the RRAMC ERASEALL operation will not automatically grant access to the debug access port.

Note: The write-buffer must be committed before initiating an erase operation.

4.2.6.4 Region protection

The RRAM memory can be divided into several regions and these regions can be configured to restrict accesses.

Each region is configured using the REGION[n].ADDRESS and REGION[n].CONFIG registers.

When the region write-once feature is enabled, writes to a 32-bit words in the region are allowed only when the current data is `0xFFFFFFFF`, else the writes are ignored.

When the region configuration registers are writable, the region configuration registers can be updated until the lock bit is set to `Enabled`. However, the register fields Read, Write, amd Execute can be written to `0`, even if the lock bit is set to `Enabled`.

After changing the region protecton, CACHE must be invalidated to stay coherent with the updated RRAM configuration. Failure to do so will cause unpredictable results, such as being able to read cached content from a memory region that was just configured to be non-readable.

Note: The configuration registers for some of the regions are read-only and are reserved by the system configurations. See the register configuration table below for regions available for users.

4.2.6.5 Immutable boot region

RRAMC can make a part of the RRAM code memory immutable.

The immutable boot region has configurable permissions settings. Read, write, and execute permissions are configured individually. By making the region read-execute only, that memory range of the RRAM becomes immutable.

The region starts at address 0x00000000 and the size of the region is configurable. Note that the region does not add additional storage, but enforces permission settings on the memory range.

The size and permission settings of the immutable boot region is configured in UICR. If UICR.BOOTCONF is not configured, RRAMC will not enforce the protection.

Once the boot region protection is enabled, it can only be removed by ERASEALL. Erase protection can be enabled to prevent ERASEALL operation. For more information about erase protection, see CTRL-AP — Control access port on page 531.

4.2.6.6 Power-failure protection

Power failure protection is possible by using the power-fail comparator (POF) that is monitoring power supply.

If the power-fail comparator is enabled, and the power supply voltage is below the POF threshold, the power-fail comparator will prevent RRAMC from performing write operations. For more information about POF, see Power-fail comparator on page 53.

If a power failure warning is present at the start of an RRAM write operation, RRAMC will block the operation and a bus error will be signaled.

If a power failure warning occurs during an ongoing RRAM write, RRAMC can be configured to handle this in two ways:

- If POWER.CONFIG.POF = `Abort`, then RRAMC will stop the commit process from the internal write-buffer as soon as the condition occurs. After a power-failure event, the write buffer must be cleared by using the CLRWRITEBUF task before writing more data to RRAM.
- If POWER.CONFIG.POF = `Wait`, then RRAMC will try to complete the on-going write despite the warning of the operating voltage becoming too low.

4.2.6.7 Registers

Instances

Instance	Domain	Base address	Description
RRAMC	GLOBAL	0x4004E000	RRAM Non-Volatile Memory Controller

Configuration

Instance	Domain	Configuration
RRAMC	GLOBAL	RRAMC.REGION[4] and RRAMC.REGION[5] is available for use, other regions are reserved for the system. RRAM word size : 64 bits per wordline Maximum write buffer size : 32

Register overview

Register	Offset	Description
TASKS_WAKEUP	0x000	Wakeup the RRAM from low power mode
TASKS_CLRWRITEBUF	0x004	Clear internal write-buffer
TASKS_COMMITWRITEBUF	0x008	Commits the data stored in internal write-buffer to RRAM
SUBSCRIBE_WAKEUP	0x080	Subscribe configuration for task WAKEUP
SUBSCRIBE_CLRWRITEBUF	0x084	Subscribe configuration for task CLRWRITEBUF
SUBSCRIBE_COMMITWRITEBUF	0x088	Subscribe configuration for task COMMITWRITEBUF
EVENTS_WOKENUP	0x100	RRAMC is woken up from low power mode
EVENTS_READY	0x104	RRAMC is ready
EVENTS_READYNEXT	0x108	Ready to accept a new write operation
EVENTS_ACCESSERROR	0x10C	RRAM access error
PUBLISH_WOKENUP	0x180	Publish configuration for event WOKENUP
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
READY	0x400	RRAMC ready status
READYNEXT	0x404	Ready next flag
ACCESSERRORADDR	0x408	Address of the first access error
BUFSTATUS.WRITEBUFEMPTY	0x418	Internal write-buffer is empty
ECC.ERRORADDR	0x420	Address of the first ECC error that could not be corrected
CONFIG	0x500	Configuration register
READYNEXTTIMEOUT	0x50C	Configuration for ready next timeout counter, in units of AXI clock frequency
POWER.CONFIG	0x510	Power configuration
POWER.LOWPOWERCONFIG	0x518	Low power mode configuration
ERASE.ERASEALL	0x540	Erase RRAM, including UICR
REGION[n].ADDRESS	0x550	Region address
REGION[n].CONFIG	0x554	Region configuration

4.2.6.7.1 TASKS_WAKEUP

Address offset: 0x000

Wakeup the RRAM from low power 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																															
A	W	TASKS_WAKEUP																																	
			Trigger	1																															

4.2.6.7.2 TASKS_CLRWRITEBUF

Address offset: 0x004

Clear internal write-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																															
A	W	TASKS_CLRWRITEBUF																																	
			Trigger	1																															

4.2.6.7.3 TASKS_COMMITWRITEBUF

Address offset: 0x008

Commits the data stored in internal write-buffer to RRAM

READY status is updated during this 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																															
A	W	TASKS_COMMITWRITEBUF																																	
			Trigger	1																															

4.2.6.7.4 SUBSCRIBE_WAKEUP

Address offset: 0x080

Subscribe configuration for task WAKEUP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task WAKEUP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.2.6.7.5 SUBSCRIBE_CLRWRITEBUF

Address offset: 0x084

Subscribe configuration for task CLRWRITEBUF

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task CLRWRITEBUF will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.2.6.7.6 SUBSCRIBE_COMMITWRITEBUF

Address offset: 0x088

Subscribe configuration for task COMMITWRITEBUF

READY status is updated during this 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				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		[0..255]	DPPI channel that task COMMITWRITEBUF will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

4.2.6.7.7 EVENTS_WOKENUP

Address offset: 0x100

RRAMC is woken up from low power mode

This event is triggered only if waken up by the WAKEUP 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_WOKENUP			RRAMC is woken up from low power mode
					This event is triggered only if waken up by the WAKEUP task
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.2.6.7.8 EVENTS_READY

Address offset: 0x104

RRAMC is ready

This event is also generated when the CPU is waking up from 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_READY			RRAMC is ready
					This event is also generated when the CPU is waking up from sleep
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.2.6.7.9 EVENTS_READYNEXT

Address offset: 0x108

Ready to accept a new write operation

This event is also generated when the CPU is waking up from 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_READYNEXT			Ready to accept a new write operation
					This event is also generated when the CPU is waking up from sleep
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.2.6.7.10 EVENTS_ACCESSERROR

Address offset: 0x10C

RRAM access 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_ACCESSERROR			RRAM access error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

4.2.6.7.11 PUBLISH_WOKENUP

Address offset: 0x180

Publish configuration for event WOKENUP

This event is triggered only if waken up by the WAKEUP task

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

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event WOKENUP will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

4.2.6.7.12 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	WOKENUP			Enable or disable interrupt for event WOKENUP This event is triggered only if waken up by the WAKEUP task
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	READY			Enable or disable interrupt for event READY This event is also generated when the CPU is waking up from sleep
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	READYNEXT			Enable or disable interrupt for event READYNEXT This event is also generated when the CPU is waking up from sleep
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ACCESSERROR			Enable or disable interrupt for event ACCESSERROR
			Disabled	0	Disable
			Enabled	1	Enable

4.2.6.7.13 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 W1S	WOKENUP			Write '1' to enable interrupt for event WOKENUP This event is triggered only if waken up by the WAKEUP task
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	READY			Write '1' to enable interrupt for event READY This event is also generated when the CPU is waking up from sleep
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	READYNEXT			Write '1' to enable interrupt for event READYNEXT This event is also generated when the CPU is waking up from sleep
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	ACCESSERROR			Write '1' to enable interrupt for event ACCESSERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.2.6.7.14 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 W1C	WOKENUP			Write '1' to disable interrupt for event WOKENUP This event is triggered only if waken up by the WAKEUP task
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	READY			Write '1' to disable interrupt for event READY This event is also generated when the CPU is waking up from sleep
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	READYNEXT			Write '1' to disable interrupt for event READYNEXT This event is also generated when the CPU is waking up from sleep
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 W1C	ACCESSERROR			Write '1' to disable interrupt for event ACCESSERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

4.2.6.7.15 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				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	WOKENUP			Read pending status of interrupt for event WOKENUP This event is triggered only if waken up by the WAKEUP task
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B	R	READY			Read pending status of interrupt for event READY This event is also generated when the CPU is waking up from sleep
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
C	R	READYNEXT			Read pending status of interrupt for event READYNEXT This event is also generated when the CPU is waking up from sleep
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
D	R	ACCESSERROR			Read pending status of interrupt for event ACCESSERROR
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

4.2.6.7.16 READY

Address offset: 0x400

RRAMC ready status

The event READY is generated when the status changes 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	R	READY			RRAMC is ready or busy The status is updated for all RRAMC operations except during read and writes to write-buffer
			Busy	0	RRAMC is busy
			Ready	1	The current RRAMC operation is completed and RRAMC is ready

4.2.6.7.17 READYNEXT

Address offset: 0x404

Ready next flag

The event READYNEXT is generated when the READYNEXT status changes 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	R	READYNEXT			RRAMC can accept a new write operation
			Busy	0	RRAMC cannot accept any write operation now
			Ready	1	RRAMC is ready to accept a new write operation

4.2.6.7.18 ACCESSERRORADDR

Address offset: 0x408

Address of the first access error

The event ACCESSERROR is generated on access error. When this event is cleared, this register is updated on the next access 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 A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00FFFFFF				**0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	ADDRESS			Access error address The most significant 8 bits are set to zero always

4.2.6.7.19 BUFSTATUS.WRITEBUFEMPTY

Address offset: 0x418

Internal write-buffer is empty

The internal write-buffer has been committed to RRAM and is now empty

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	EMPTY			
			NotEmpty	0	The internal write-buffer has data that needs committing
			Empty	1	The internal write-buffer is empty and has no content that needs to be committed

4.2.6.7.20 ECC.ERRORADDR

Address offset: 0x420

Address of the first ECC error that could not be corrected

The event ECCERROR is generated on ECC error. When this event is cleared, this register is updated on the next ECC 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00FFFFFF				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	R/W	Field	Value ID	Value																															
A	R	ADDRESS																																	
				ECC error address The most significant 8 bits are set to zero always																															

4.2.6.7.21 CONFIG

Address offset: 0x500

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	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																															
A	RW	WEN																																	
			Disabled	0																															
			Enabled	1																															
B	RW	WRITEBUFSIZE		0..8																															
			Unbuffered	0																															

4.2.6.7.22 READYNEXTTIMEOUT

Address offset: 0x50C

Configuration for ready next timeout counter, in units of AXI clock frequency

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				B																				A	A	A	A	A	A	A	A	A	A	A	A
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**	**0**
ID	R/W	Field	Value ID	Value																															
A	RW	VALUE		[0..4095]																															
B	RW	EN																																	
				Enable ready next timeout The timeout value is number of RRAMC clock cycles. The timeout starts when the READYNEXT is set to ready																																	
			Disable	0																															
			Enable	1																															

4.2.6.7.23 POWER.CONFIG

Address offset: 0x510

Power 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 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**
ID	R/W	Field	Value ID	Value	Description
A	RW	ACCESSTIMEOUT			Access timeout, in 31.25 ns units, used for going into standby power mode or remain active on wake up The timeout counter counts down and is restarted on every RRAM access and on event WOKENUP
B	RW	POF			Power on failure warning handling configuration
			Wait	0	Wait until the current RRAM write finishes
			Abort	1	Abort the current RRAM write

4.2.6.7.24 POWER.LOWPOWERCONFIG

Address offset: 0x518

Low power mode configuration

The RRAMC low power mode is entered while the device goes into system on 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			RRAM low power mode
			PowerDown	0	The RRAM goes into power down mode
			Standby	1	The RRAM automatically goes into standby mode while the RRAM is not being accessed This mode gives faster wake-ups, and is useful in combination with Constant Latency sub-power mode.
			NAP	2	The RRAM goes into NAP mode
			PowerOff	3	The RRAM is powered Off

4.2.6.7.25 ERASE.ERASEALL

Address offset: 0x540

Erase RRAM, including UICR

The status in READY is updated during this operation

Writes to this register are ignored when erase protect 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	RW	ERASE			Erase RRAM
			NoOperation	0	No operation
			Erase	1	Start erase of chip

4.2.6.7.26 REGION[n] (n=0..4)

RRAMC can apply access privileges to regions of the RRAM. Some regions are dedicated for system use and are not available for configuration - refer to the instantiation table for details.

4.2.6.7.26.1 REGION[n].ADDRESS (n=0..4)

Address offset: 0x550 + (n × 0x8)

Region 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																															
A	RW	STARTADDR																																	

Start address of the region [n]

Bits [31:24] and bit [9:0] are read-only and set to zero

4.2.6.7.26.2 REGION[n].CONFIG (n=0..4)

Address offset: 0x554 + (n × 0x8)

Region configuration

The register fields READ, WRITE and EXECUTE can can be written to 0, even when the LOCK field is set to 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	F	F	F	F	F	F	F	F	F			E	D									C	B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	READ																																	
			NotAllowed	0																															
			Allowed	1																															
B	RW	WRITE																																	
			NotAllowed	0																															
			Allowed	1																															
C	RW	EXECUTE																																	
			NotAllowed	0																															
			Allowed	1																															
D	RW	WRITEONCE																																	
			Disabled	0																															
			Enabled	1																															
E	RW	LOCK																																	
	W1S																																		
			Disabled	0																															
			Enabled	1																															
F	RW	SIZE																																	

Read access

Read access to override region [n] is not allowed

Read access to override region [n] is allowed

Write access

Write access to override region [n] is not allowed

Write access to override region [n] is allowed

Execute access

Execute access to override region [n] is not allowed

Execute access to override region [n] is allowed

Write-once

Write-once disabled

Write-once enabled

Enable lock

Lock disabled for region [n]

Lock enabled for region [n]

Size in KBytes of region [n]

4.2.7 SWI — Software interrupts

A set of interrupts have been reserved for use as software interrupts.

These interrupts can be enabled and triggered by software by using the Arm Cortex-M33 NVIC registers, as described in the *Arm Cortex-M33 Processor Technical Reference Manual*.

4.2.7.1 Registers

Instances

Instance	Domain	Base address	Description
SWI00	APPLICATION	0x4001C000	Software interrupt SWI00
SWI01	APPLICATION	0x4001D000	Software interrupt SWI01
SWI02	APPLICATION	0x4001E000	Software interrupt SWI02
SWI03	APPLICATION	0x4001F000	Software interrupt SWI03

4.2.8 UICR — User information configuration registers

The user information configuration registers (UICR) are non-volatile memory (NVM) registers that configure user specific settings and values for emulated one-time programmable (OTP).

All UICR registers have a RW1 protection, which means that they can be read multiple times, but written only once when UICR has been erased by the Erase All operation.

For information on writing registers, see RRAMC — Resistive random access memory controller on page 32 and Memory on page 11.

Notice that all access port protection registers are duplicated into PROTECT0/PROTECT1. For optimal security, set both registers set to "random" values different from the Unprotected value. For ERASEPROTECT, set both PROTECT0/PROTECT1 registers to the Protected value.

4.2.8.1 Registers

Instances

Instance	Domain	Base address	Description
UICR	GLOBAL	0x00FFD000	User information configuration

Register overview

Register	Offset	Description
APPROTECT[n].PROTECT0	0x000	Access port protection
APPROTECT[n].PROTECT1	0x00C	Access port protection
ERASEPROTECT[n].PROTECT0	0x60	Erase protection
ERASEPROTECT[n].PROTECT1	0x6C	Erase protection
BOOTCONF	0x080	Immutable boot region configuration.
USER.ROT.PUBKEY[n].DIGEST[o]	0x200	First 256 bits of SHA2-512 digest over RoT public key generation [n].
USER.ROT.PUBKEY[n].REVOKE[o]	0x220	Revocation status for RoT public key generation [n].
USER.ROT.AUTHOPKEY[n].DIGEST[o]	0x2B0	First 256 bits of SHA2-512 digest over RoT authenticated operation public key generation [n].
USER.ROT.AUTHOPKEY[n].REVOKE[o]	0x2D0	Revocation status for RoT authenticated operation public key generation [n].
OTP[n]	0x500	One time programmable memory

4.2.8.1.1 APPROTECT[n] (n=0..0)

Access Port Protection Registers

4.2.8.1.1.1 APPROTECT[n].PROTECT0 (n=0..0)

Address offset: 0x000 + (n × 0x10)

Access port protection

Any other value than Unprotected will lock TAMPC PROTECT.DOMAIN signal protectors.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	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	RW1	PALL																																	
			Unprotected	0xFFFFFFFF																															

Leaves TAMPC PROTECT.DOMAIN DBGEN and NIDEN signal protectors unlocked and under CPU control.

4.2.8.1.1.2 APPROTECT[n].PROTECT1 (n=0..0)

Address offset: 0x00C + (n × 0x10)

Access port protection

Any other value than Unprotected will lock TAMPC PROTECT.DOMAIN signal protectors.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	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	RW1	PALL																																	
			Unprotected	0xFFFFFFFF																															

Leaves TAMPC PROTECT.DOMAIN DBGEN and NIDEN signal protectors unlocked and under CPU control.

4.2.8.1.2 ERASEPROTECT[n] (n=0..0)

Erase Protection Registers

4.2.8.1.2.1 ERASEPROTECT[n].PROTECT0 (n=0..0)

Address offset: 0x60 + (n × 0x10)

Erase protection

Any other value than `Protected` will leave the TAMPC PROTECT.ERASEPROTECT signal protector unlocked, so that CPU can control its 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																															
A	RW1	PALL																																	
			Protected	0x50FA50FA																															

Erase protection is enabled and the signal protector is locked.

4.2.8.1.2.2 ERASEPROTECT[n].PROTECT1 (n=0..0)

Address offset: 0x6C + (n × 0x10)

Erase protection

Any other value than `Protected` will leave the TAMPC PROTECT.ERASEPROTECT signal protector unlocked, so that CPU can control its 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																															
A	RW1	PALL																																	
			Protected	0x50FA50FA																															
				Erase protection is enabled and the signal protector is locked.																															

4.2.8.1.3 BOOTCONF

Address offset: 0x080

Immutable boot region configuration.

If this register is not equal to 0xFFFFFFFF, RRAMC applies these settings to form the immutable boot region.

For an immutable bootloader, recommended value is READ, EXECUTE, WRITEONCE, LOCK, and SIZE. READ permission is needed for the CPU to read constants and instructions.

Unused bits (unused fields) must be set to zero.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID										F	F	F	F	F	F	F	F	F	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																															
A	RW1	READ																																	
				Read access. Must be enabled in order for the Arm Cortex CPU to start executing from RRAM.																																	
			NotAllowed	0																															
				Reading from the region is not allowed.																															
			Allowed	1																															
				Reading from the region is allowed																															
B	RW1	WRITE																																	
				Write access																															
			NotAllowed	0																															
				Writing to the region is not allowed																															
			Allowed	1																															
				Writing to the region is allowed																															
C	RW1	EXECUTE																																	
				Execute access																															
			NotAllowed	0																															
				Executing code from the region is not allowed																															
			Allowed	1																															
				Executing code from the region is allowed																															
D	RW1	WRITEONCE																																	
				Write-once																															
			Disabled	0																															
				Write-once disabled																															
			Enabled	1																															
				Write-once enabled Writes to a 32-bit word in the BOOTCONF region are is only when the current data is 0xFFFFFFFF, otherwise the writes are ignored																															
E	RW1	LOCK																																	
				Enable lock of configuration register																															
			Disabled	0																															
				Lock is disabled, and the RRAMC region configuration registers for the immutable boot region are writable.																															
			Enabled	1																															
				Lock is enabled, and the RRAMC configuration registers for the immutable boot region are read-only.																															
F	RW1	SIZE																																	
				Immutable boot region size Configures the region size in kB																															

4.2.8.1.4 USER.ROT

Assets installed to establish initial Root of Trust in the device.

User RoT key materials

4.2.8.1.4.1 USER.ROT.PUBKEY[n].DIGEST[o] (n=0..3) (o=0..7)

Address offset: 0x200 + (n × 0x2C) + (o × 0x4)

First 256 bits of SHA2-512 digest over RoT public key generation [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	RW1	VALUE			Value for word [o] in the key digest [n].

4.2.8.1.4.2 USER.ROT.PUBKEY[n].REVOKE[o] (n=0..3) (o=0..2)

Address offset: 0x220 + (n × 0x2C) + (o × 0x4)

Revocation status for RoT public key generation [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	RW1	STATUS			Revocation status.
			NotRevoked	0xFFFFFFFF	Key not revoked.
					Any other value says the key is revoked.

4.2.8.1.4.3 USER.ROT.AUTHOPKEY[n].DIGEST[o] (n=0..3) (o=0..7)

Address offset: 0x2B0 + (n × 0x2C) + (o × 0x4)

First 256 bits of SHA2-512 digest over RoT authenticated operation public key generation [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	RW1	VALUE			Value for word [o] in the key digest [n].

4.2.8.1.4.4 USER.ROT.AUTHOPKEY[n].REVOKE[o] (n=0..3) (o=0..2)

Address offset: 0x2D0 + (n × 0x2C) + (o × 0x4)

Revocation status for RoT authenticated operation public key generation [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	RW1	STATUS			Revocation status.
			NotRevoked	0xFFFFFFFF	Key not revoked.
					Any other value says the key is revoked.

4.2.8.1.5 OTP[n] (n=0..319)

Address offset: 0x500 + (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	A	A	A	A	A	A	A	A	A	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	RW1	OTP			OTP word Can only be written to a non 0xFFFFFFFF value once after Erase All operation.

5 Power and clock management

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

The power and clock management system is shown in the following figure.

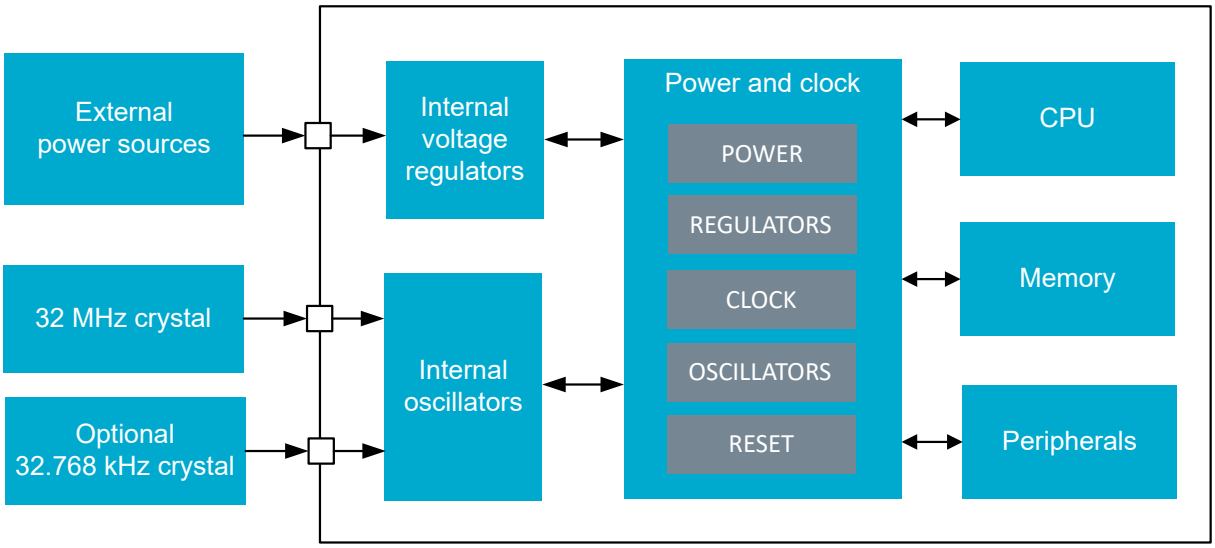


Figure 8: Power and clock management

The power and clock management system automatically tracks the power and clock resources requested by components in the system at any given time. To achieve the lowest power consumption possible, the system evaluates the requests, starts and stops clock sources, and chooses the most optimal regulator operation modes.

The device start-up sequence after reset is described in RESET — Reset control on page 88.

5.1 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 power and clock management unit can switch the appropriate internal power domains on and off, depending on power requirements. A peripheral's power requirement is directly related to its activity level, which increases and decreases when specific tasks are triggered or events are generated.

5.1.1 Sub-power modes

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

The sub-power modes are the following:

- Constant Latency
- Low-power

In Constant Latency mode, the CPU wakeup latency and the PPI task response are constant and kept at a minimum. This is secured by forcing a set of basic resources to be turned on while in sleep mode. Constant

and predictable latency increases power consumption. Constant Latency mode is selected by triggering the task CONSTLAT.

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

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

5.2 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 most ongoing tasks are stopped.

Register SYSTEMOFF on page 86 sets the device into System OFF mode. The following wakeup sources will initiate a wakeup from System OFF:

- The DETECT signal generated by the GPIO peripheral
- The SYSCOUNTER compare event generated by the GRTC peripheral
- 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 Reset behavior on page 90.

One or more RAM sections can be retained in System OFF depending on the RAM retention settings configured in MEMCONF — Memory configuration on page 30.

Before entering System OFF mode, the following conditions must be met.

- All on-going EasyDMA transactions must finish. See peripheral specific chapters for more information about how to get the status of EasyDMA transactions.
- The 32 MHz oscillator (HFXO) must be stopped. Stop HFXO with the XOSTOP task. The 32 kHz oscillator (LFXO) can be running.
- The register RESET.RESETREAS must be cleared. Failure to do so can make the system immediately wake up from System OFF mode.

5.2.1 Emulated System OFF mode

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

Resources required for debugging include the following key components:

- Debug Interface mode on page 530
- CLOCK — Clock control on page 54
- POWER — Power control on page 79
- OSCILLATORS — Oscillator control on page 73
- REGULATORS — Regulator control on page 84
- RESET — Reset control on page 88
- CPU on page 16
- Memory on page 11

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. This prevents the CPU from executing code that normally should not be executed. For more information, see Debug on page 526.

5.3 Power supply supervisors

The power supply supervisors monitor the connected power supply and provide the following functionality.

- Power-on reset (POR) — signals the circuit when a supply is connected
- Fixed brownout reset detector (BOR) — holds the system in reset when the voltage is too low for safe operation
- Optional power-fail comparator (POF) — signals the application when the supply voltage drops below a configured threshold

The power supply supervisors are illustrated in the following figure.

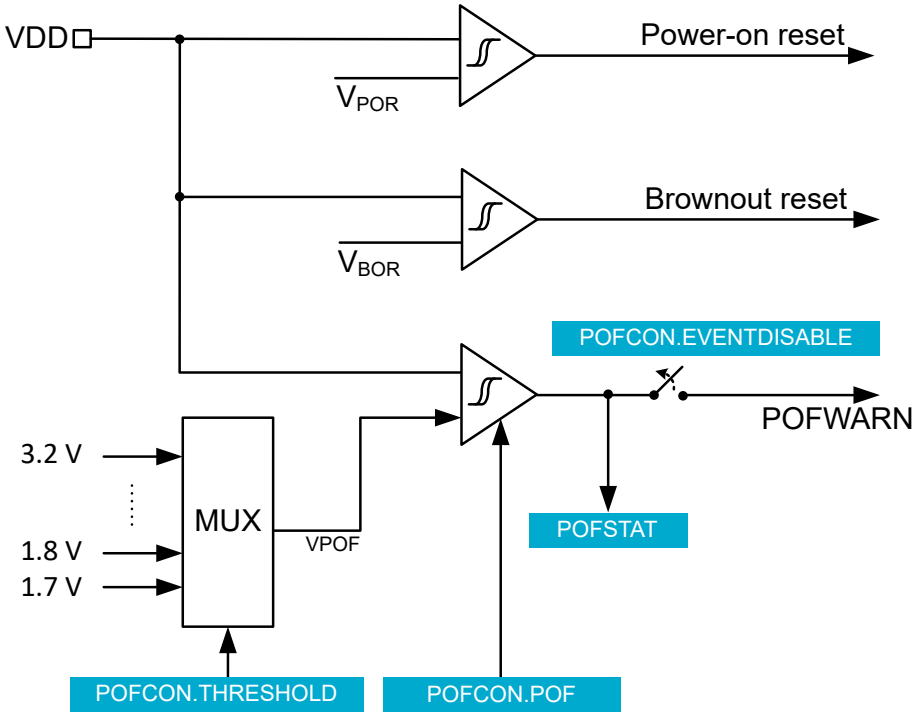


Figure 9: Power supply supervisors

5.3.1 Power-fail comparator

Using the power-fail comparator (POF) is optional. When enabled, it notifies the CPU of a potential power supply failure.

The POF can measure the voltage on **VDD** . To enable and configure the POF, see register POFCON (Retained) on page 86.

When the supply voltage falls below the defined threshold, the POF generates an event POFWARN that can be used by an application to prepare for power failure. This event is also generated when the supply voltage is already below the threshold at the time the POF is enabled, or if the threshold is reconfigured to a level above the supply voltage. POFWARN is disabled using the EVENTDISABLE field of REGULATORS.POFCON. In addition to the event, the result of the POF is found using POFSTAT on page 87.

POFWARN prevents RRAMC from performing write operations to the non-volatile memory. See RRAMC — Resistive random access memory controller on page 32 for more information about non-volatile memory.

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

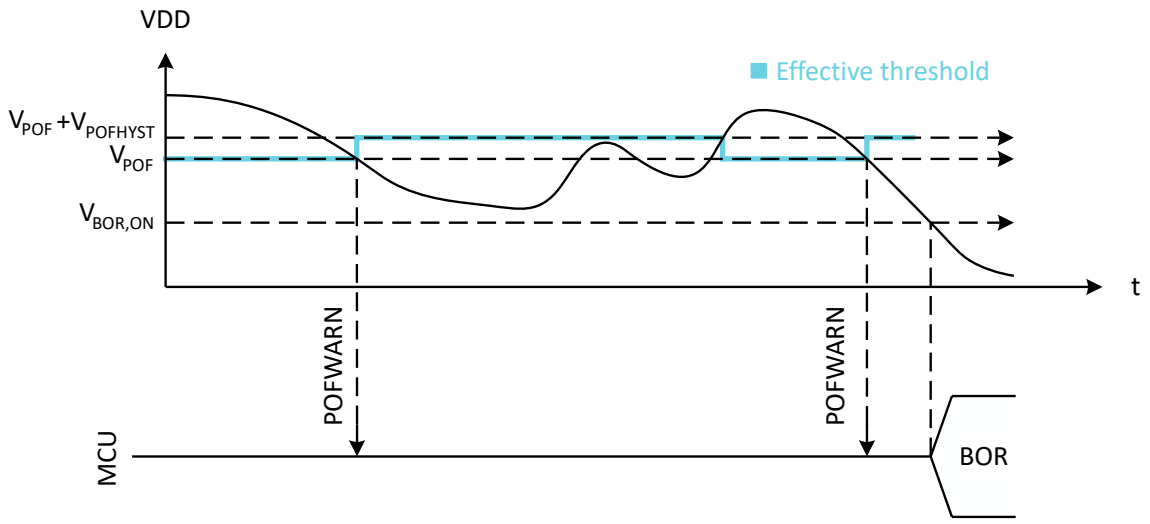


Figure 10: Power-fail comparator

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

To measure the voltage, perform the following steps.

1. Disable POFWARN by writing `Disabled` to REGULATORS.POFCON.EVENTDISABLE.
2. Enable POF by writing `Enabled` to REGULATORS.POFCON.POF.
3. Loop over all threshold voltages by writing a threshold voltage into register REGULATORS.POFCON.THRESHOLD, starting at the lowest value enumerator until REGULATORS.POFSTAT toggles. This toggle indicates that the voltage has been found and can be read from register REGULATORS.POFCON.THRESHOLD.

5.4 CLOCK — Clock control

The clock control system sources the system clocks from internal or external high and low frequency oscillators. It distributes the clocks to modules based on their requirements. Clock distribution is automated and grouped independently by module to limit current consumption in unused branches of the clock tree.

The following are the main features for CLOCK:

- On-chip 128 MHz phase-locked loop (PLL) with internal oscillator
- 32 MHz crystal oscillator (when using the external 32 MHz crystal)
- 32.768 kHz RC oscillator
- 32.768 kHz crystal oscillator (when using the external 32.768 kHz crystal)
- Automatic clock control and distribution

The clock control system is responsible for requesting resources from the power and clock subsystem.

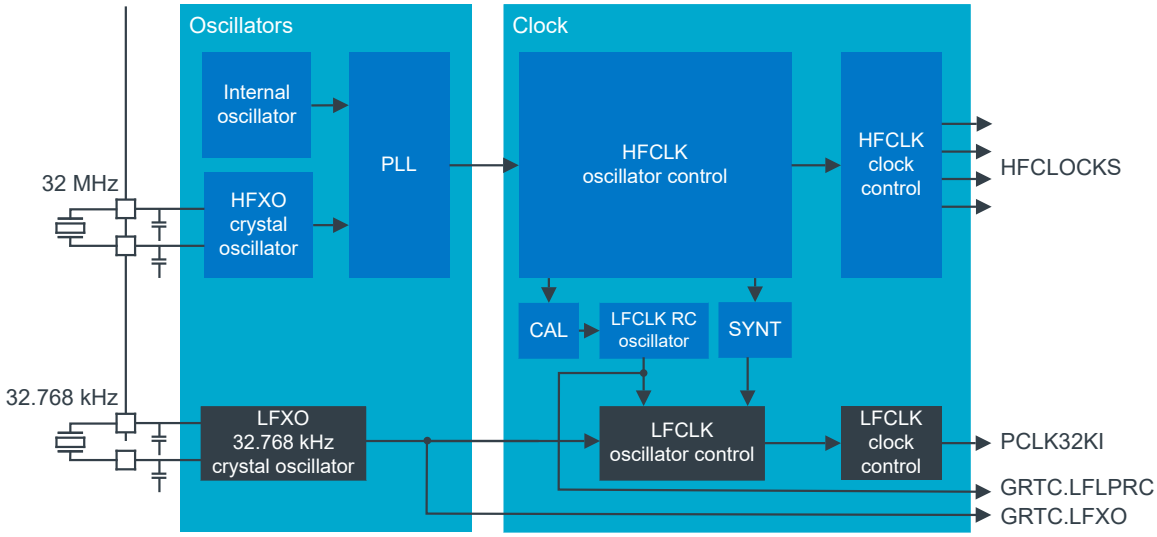


Figure 11: Clock control

5.4.1 HFCLK controller

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

Clock	Description
HCLKCORE	MCU power domain and CPU clock where 64 or 128 MHz can be selected
PCLK32M	32 MHz peripheral clock
PCLK16M	16 MHz peripheral clock
PCLK1M	1 MHz peripheral clock

Table 17: Clocks

Clock source	Description
HFINT	128 MHz internal oscillator
HFXO	32 MHz crystal oscillator

Table 18: Sources

The following HFCLK sources generate the HFCLK clocks:

- 128 MHz internal oscillator — PLL is operating in free running mode
- 32 MHz crystal oscillator — PLL is locked on a crystal (XOSC), optionally using built-in capacitors as described in OSCILLATORS — Oscillator control on page 73.

CPUs, peripherals, and other system components automatically request clocks. The HFCLK control passes the request to the power and clock subsystem. When the clocks are running, the HFCLK control distributes them to the components. The CPU clock frequency can be selected, as described in OSCILLATORS — Oscillator control on page 73.

When all HFCLK control requests end, the HFCLK control stops requesting CLOCK from the power and clock subsystem. For example, when the CPU enters sleep or when peripherals have completed their tasks, HFCLK stops CLOCK requests. If there are no requests for HFCLK or PCLK control, the power and clock subsystem automatically stops the clock.

When the system enters System ON mode and an HFCLK clock is requested, the PLL is automatically started. When clock requests stop, the PLL automatically stops.

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

An HFCLK source can run before being started by the relevant clock request. This reduces start-up time but causes increased power consumption. An example of this would be to keep the PLL running during sleep by using the task PLLSTART.

HFXO must be started when crystal clock accuracy is required. The crystal is started by triggering the task XOSTART. When the crystal reaches the correct amplitude and frequency, the PLL automatically locks to the crystal and generates the event XOSTARTED. At the same time, the crystal oscillator is performing an XOTUNE. When that process finishes, the event XOTUNED is generated indicating the signal from the crystal oscillator is accurately tuned. The XOTUNED event is always generated after the XOSTARTED event. If the XOTUNE process fails, the event XOTUNEFAILED is generated. When that happens, the XOTUNE process must be started by triggering the task XOTUNE. The task XOSTOP will stop the crystal oscillator and the XOTUNE process (if it is running). The XOTUNE process will continue from where it left off, when the XOSTART task is triggered again.

If the crystal oscillator requires the XOTUNE process to be repeated, the device generates the event XOTUNEERROR. When that happens, the XOTUNE task must be triggered. This task does not request the crystal oscillator to start and it only takes effect if the crystal oscillator is already running. Do not trigger this task at the same time that RADIO is running (meaning RADIO must not be in the RX or TX states). A successful XOTUNE process is indicated by the event XOTUNED. If the procedure fails to complete, the XOTUNEFAILED event is generated. When a failure is detected, the XOTUNE process must be repeated. Repeated failures to complete the XOTUNED process indicates a faulty crystal or load capacitors.

Note: The crystal oscillator quality indicated by the XOSTARTED event is sufficient for all peripherals except RADIO and when calibrating the 32.768 kHz oscillator. Before using RADIO, ensure that the event XOTUNED has been generated. This ensures the highest quality crystal signal is available.

A new START task can be initiated after one has already been triggered, but 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 tasks is generated, generates a new STARTED event.

The amount of time between a START task and its 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. Different crystal types also have different start-up times, see OSCILLATORS — Oscillator control on page 73 for details.

HFXO must be running to use RADIO, UARTE, or to calibrate the 32.768 kHz RC oscillator. Using HFXO will also improve SAADC performance by reducing clock jitter. When using serial communication peripherals such as SPIM, SPIS, TWIM, and TWIS, the HFXO must be running to achieve the highest accuracy for the bit rate. When using the internal RC oscillator (HFINT), the frequency accuracy of the serial interface is limited to the accuracy of HFINT, see High frequency clock source (HFCLK) on page 554.

5.4.2 LFCLK controller

The system supports the following low frequency clock sources, as described in Clock control on page 55.

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

LFXO can run in System OFF mode. The other clock sources only run in System ON mode.

The following LF clocks are available in the system.

Clock	Description
PCLK32KI	32.768 kHz peripheral low-frequency clock.
GRTC.LFLPRC	Direct path from 32.768 kHz internal oscillator (LFRC) to GRTC peripheral. Available only in System ON mode. If GRTC.LFLPRC is used, stop GRTC before entering System OFF mode.
GRTC.LFXO	Direct path from 32.768 kHz crystal oscillator (LFXO) to GRTC peripheral. Available in System ON or System OFF mode.

Table 19: Clocks

When a peripheral requires the PCLK32KI clock, the LFCLK control automatically requests the LFCLK clock for the power and clock subsystem. The default LFCLK source is the LFRC.

To use a different LFCLK source, select the preferred clock source in register LFCLK.SRC on page 71 and then trigger the LFCLKSTART task. If LFXO is selected as the clock source, LFCLK initially starts running from the 32.768 kHz LFRC then automatically switches to the crystal once available. The LFCLKSTARTED event is then generated.

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

The LFCLKSTART task will request the clock to keep running until triggering the LFCLKSTOP task to stop the clock.

The LFCLK clock is stopped when there are no requests. For example, WDT is stopped, and the LFCLKSTOP task is triggered. Triggering the LFCLKSTOP task is required after the LFCLKSTART task has been triggered.

5.4.2.1 Calibrating the 32.768 kHz RC oscillator

The LFRC frequency is affected by temperature variation. LFRC can be calibrated to improve accuracy by using HFCLK as a reference oscillator during calibration.

The calibration must use the following sequence.

1. Start the LFCLK by triggering the LFCLKSTART task.
2. Start the HFCLK crystal oscillator HFXO by triggering the XOSTART task.
3. Wait for the LFCLKSTARTED and the HFXO XOTUNED events.
4. Trigger the CAL task to start the calibration process. The device automatically performs the calibration, adjusting the LFCLK frequency using HFCLK as reference. The DONE event is generated when calibration finishes.
5. Stop HFXO with the XOSTOP task.
6. Stop LFCLK with the LFCLKSTOP task.

LFCLK uses the calibrated value until the next calibration.

5.4.3 Registers

Instances

Instance	Domain	Base address	Description
CLOCK	GLOBAL	0x4010E000	Clock control

Configuration

Instance	Domain	Configuration
CLOCK	GLOBAL	

Register overview

Register	Offset	Description
TASKS_XOSTART	0x000	Start crystal oscillator (HFXO)
TASKS_XOSTOP	0x004	Stop crystal oscillator (HFXO)
TASKS_PLLSTART	0x008	Start PLL and keep it running, regardless of the automatic clock requests
TASKS_PLLSTOP	0x00C	Stop PLL
TASKS_LFCLKSTART	0x010	Start LFCLK source as selected in LFCLK.SRC
TASKS_LFCLKSTOP	0x014	Stop LFCLK source
TASKS_CAL	0x018	Start calibration of LFRC oscillator
TASKS_XOTUNE	0x01C	Request tuning for HFXO
TASKS_XOTUNEABORT	0x020	Abort tuning for HFXO
SUBSCRIBE_XOSTART	0x080	Subscribe configuration for task XOSTART
SUBSCRIBE_XOSTOP	0x084	Subscribe configuration for task XOSTOP
SUBSCRIBE_PLLSTART	0x088	Subscribe configuration for task PLLSTART
SUBSCRIBE_PLLSTOP	0x08C	Subscribe configuration for task PLLSTOP
SUBSCRIBE_LFCLKSTART	0x090	Subscribe configuration for task LFCLKSTART
SUBSCRIBE_LFCLKSTOP	0x094	Subscribe configuration for task LFCLKSTOP
SUBSCRIBE_CAL	0x098	Subscribe configuration for task CAL
SUBSCRIBE_XOTUNE	0x09C	Subscribe configuration for task XOTUNE
SUBSCRIBE_XOTUNEABORT	0x0A0	Subscribe configuration for task XOTUNEABORT
EVENTS_XOSTARTED	0x100	Crystal oscillator has started
EVENTS_PLLSTARTED	0x104	PLL started
EVENTS_LFCLKSTARTED	0x108	LFCLK source started
EVENTS_DONE	0x10C	Calibration of LFRC oscillator complete event
EVENTS_XOTUNED	0x110	HFXO tuning is done. XOTUNED is generated after TASKS_XOSTART or after TASKS_XOTUNE has completed
EVENTS_XOTUNEERROR	0x114	HFXO quality issue detected, XOTUNE is needed
EVENTS_XOTUNEFAILED	0x118	HFXO tuning could not be completed
PUBLISH_XOSTARTED	0x180	Publish configuration for event XOSTARTED
PUBLISH_PLLSTARTED	0x184	Publish configuration for event PLLSTARTED
PUBLISH_LFCLKSTARTED	0x188	Publish configuration for event LFCLKSTARTED
PUBLISH_DONE	0x18C	Publish configuration for event DONE
PUBLISH_XOTUNED	0x190	Publish configuration for event XOTUNED
PUBLISH_XOTUNEERROR	0x194	Publish configuration for event XOTUNEERROR
PUBLISH_XOTUNEFAILED	0x198	Publish configuration for event XOTUNEFAILED
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
XO.RUN	0x408	Indicates that XOSTART task was triggered
XO.STAT	0x40C	XO status
PLL.RUN	0x428	Indicates that PLLSTART task was triggered
PLL.STAT	0x42C	Which PLL settings were selected when triggering START task
LFCLK.SRC	0x440	Clock source for LFCLK
LFCLK.RUN	0x448	Indicates that LFCLKSTART task was triggered
LFCLK.STAT	0x44C	Copy of LFCLK.SRCCOPY register, set when LFCLKSTARTED event is triggered.
LFCLK.SRCCOPY	0x450	Copy of LFCLK.SRC register, set when LFCLKSTART task is triggered

5.4.3.1 TASKS_XOSTART

Address offset: 0x000

Start crystal oscillator (HFXO)

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

5.4.3.2 TASKS_XOSTOP

Address offset: 0x004

Stop crystal oscillator (HFXO)

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

5.4.3.3 TASKS_PLLSTART

Address offset: 0x008

Start PLL and keep it running, regardless of the automatic clock requests

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_PLLSTART			Start PLL and keep it running, regardless of the automatic clock requests
			Trigger	1	Trigger task

5.4.3.4 TASKS_PLLSTOP

Address offset: 0x00C

Stop PLL

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

5.4.3.5 TASKS_LFCLKSTART

Address offset: 0x010

Start LFCLK source as selected in LFCLK.SRC

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

5.4.3.6 TASKS_LFCLKSTOP

Address offset: 0x014

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

5.4.3.7 TASKS_CAL

Address offset: 0x018

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

5.4.3.8 TASKS_XOTUNE

Address offset: 0x01C

Request tuning for HFXO

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

5.4.3.9 TASKS_XOTUNEABORT

Address offset: 0x020

Abort tuning for HFXO

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

5.4.3.10 SUBSCRIBE_XOSTART

Address offset: 0x080

Subscribe configuration for task XOSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task XOSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.11 SUBSCRIBE_XOSTOP

Address offset: 0x084

Subscribe configuration for task XOSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task XOSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.12 SUBSCRIBE_PLLSTART

Address offset: 0x088

Subscribe configuration for task PLLSTART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task PLLSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.13 SUBSCRIBE_PLLSTOP

Address offset: 0x08C

Subscribe configuration for task PLLSTOP

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task PLLSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.14 SUBSCRIBE_LFCLKSTART

Address offset: 0x090

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		[0..255]	DPPI channel that task LFCLKSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.15 SUBSCRIBE_LFCLKSTOP

Address offset: 0x094

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		[0..255]	DPPI channel that task LFCLKSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.16 SUBSCRIBE_CAL

Address offset: 0x098

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		[0..255]	DPPI channel that task CAL will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.17 SUBSCRIBE_XOTUNE

Address offset: 0x09C

Subscribe configuration for task XOTUNE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task XOTUNE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.18 SUBSCRIBE_XOTUNEABORT

Address offset: 0x0A0

Subscribe configuration for task XOTUNEABORT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task XOTUNEABORT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.4.3.19 EVENTS_XOSTARTED

Address offset: 0x100

Crystal oscillator 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_XOSTARTED			Crystal oscillator has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

5.4.3.20 EVENTS_PLLSTARTED

Address offset: 0x104

PLL 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_PLLSTARTED			PLL started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

5.4.3.21 EVENTS_LFCLKSTARTED

Address offset: 0x108

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

5.4.3.22 EVENTS_DONE

Address offset: 0x10C

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

5.4.3.23 EVENTS_XOTUNED

Address offset: 0x110

HFXO tuning is done. XOTUNED is generated after TASKS_XOSTART or after TASKS_XOTUNE has 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_XOTUNED			HFXO tuning is done. XOTUNED is generated after TASKS_XOSTART or after TASKS_XOTUNE has completed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

5.4.3.24 EVENTS_XOTUNEERROR

Address offset: 0x114

HFXO quality issue detected, XOTUNE is needed

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_XOTUNEERROR			HFXO quality issue detected, XOTUNE is needed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

5.4.3.25 EVENTS_XOTUNEFAILED

Address offset: 0x118

HFXO tuning could not be 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_XOTUNEFAILED			HFXO tuning could not be completed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

5.4.3.26 PUBLISH_XOSTARTED

Address offset: 0x180

Publish configuration for event XOSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event XOSTARTED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.27 PUBLISH_PLLSTARTED

Address offset: 0x184

Publish configuration for event PLLSTARTED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event PLLSTARTED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.28 PUBLISH_LFCLKSTARTED

Address offset: 0x188

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event LFCLKSTARTED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.29 PUBLISH_DONE

Address offset: 0x18C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DONE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.30 PUBLISH_XOTUNED

Address offset: 0x190

Publish configuration for event XOTUNED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event XOTUNED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.31 PUBLISH_XOTUNEERROR

Address offset: 0x194

Publish configuration for event XOTUNEERROR

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event XOTUNEERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.32 PUBLISH_XOTUNEFAILED

Address offset: 0x198

Publish configuration for event XOTUNEFAILED

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event XOTUNEFAILED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.4.3.33 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				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	XOSTARTED			Enable or disable interrupt for event XOSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	PLLSTARTED			Enable or disable interrupt for event PLLSTARTED
			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				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	LFCLKSTARTED			Enable or disable interrupt for event LFCLKSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	DONE			Enable or disable interrupt for event DONE
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	XOTUNED			Enable or disable interrupt for event XOTUNED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	XOTUNEERROR			Enable or disable interrupt for event XOTUNEERROR
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	XOTUNEFAILED			Enable or disable interrupt for event XOTUNEFAILED
			Disabled	0	Disable
			Enabled	1	Enable

5.4.3.34 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				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 W1S	XOSTARTED			Write '1' to enable interrupt for event XOSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	PLLSTARTED			Write '1' to enable interrupt for event PLLSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	LFCLKSTARTED			Write '1' to enable interrupt for event LFCLKSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	DONE			Write '1' to enable interrupt for event DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	XOTUNED			Write '1' to enable interrupt for event XOTUNED
			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				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
F	RW W1S	XOTUNEERROR			Write '1' to enable interrupt for event XOTUNEERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	XOTUNEFAILED			Write '1' to enable interrupt for event XOTUNEFAILED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.4.3.35 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				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 W1C	XOSTARTED			Write '1' to disable interrupt for event XOSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	PLLSTARTED			Write '1' to disable interrupt for event PLLSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	LFCLKSTARTED			Write '1' to disable interrupt for event LFCLKSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	DONE			Write '1' to disable interrupt for event DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	XOTUNED			Write '1' to disable interrupt for event XOTUNED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	XOTUNEERROR			Write '1' to disable interrupt for event XOTUNEERROR
			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				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
G	RW W1C	XOTUNEFAILED			Write '1' to disable interrupt for event XOTUNEFAILED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.4.3.36 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				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	R	XOSTARTED			Read pending status of interrupt for event XOSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B	R	PLLSTARTED			Read pending status of interrupt for event PLLSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
C	R	LFCLKSTARTED			Read pending status of interrupt for event LFCLKSTARTED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
D	R	DONE			Read pending status of interrupt for event DONE
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
E	R	XOTUNED			Read pending status of interrupt for event XOTUNED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
F	R	XOTUNEERROR			Read pending status of interrupt for event XOTUNEERROR
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
G	R	XOTUNEFAILED			Read pending status of interrupt for event XOTUNEFAILED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

5.4.3.37 XO.RUN

Address offset: 0x408

Indicates that XOSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			XOSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

5.4.3.38 XO.STAT

Address offset: 0x40C

XO 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	STATE			XO state (Running between START task and STOPPED event)
			NotRunning	0	XO is not running
			Running	1	XO is running

5.4.3.39 PLL.RUN

Address offset: 0x428

Indicates that PLLSTART 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			PLLSTART task triggered or not
			NotTriggered	0	Task not triggered
			Triggered	1	Task triggered

5.4.3.40 PLL.STAT

Address offset: 0x42C

Which PLL settings were selected when triggering START task

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			PLL state (Running between START task and STOPPED event)
			NotRunning	0	PLL is not running
			Running	1	PLL is running

5.4.3.41 LFCLK.SRC

Address offset: 0x440

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 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Select which LFCLK source is started by the LFCLKSTART task
			LFRC	0	32.768 kHz RC oscillator
			LFXO	1	32.768 kHz crystal oscillator
			LFSYNT	2	32.768 kHz synthesized from HFCLK

5.4.3.42 LFCLK.RUN

Address offset: 0x448

Indicates that 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

5.4.3.43 LFCLK.STAT

Address offset: 0x44C

Copy of LFCLK.SRCCOPY register, set when LFCLKSTARTED 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Value of LFCLK.SRCCOPY register when LFCLKSTARTED event was triggered
			LFRC	0	32.768 kHz RC oscillator
			LFXO	1	32.768 kHz crystal oscillator
			LFSYNT	2	32.768 kHz synthesized from HFCLK
B	R	STATE			LFCLK state (Running between START task and STOPPED event)
			NotRunning	0	LFCLK not running
			Running	1	LFCLK running

5.4.3.44 LFCLK.SRCCOPY

Address offset: 0x450

Copy of LFCLK.SRC register, set when LFCLKSTART 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	SRC			Value of LFCLK.SRC register when LFCLKSTART task was triggered
			LFRC	0	32.768 kHz RC oscillator
			LFXO	1	32.768 kHz crystal oscillator
			LFSYNT	2	32.768 kHz synthesized from HFCLK

5.5 OSCILLATORS — Oscillator control

The system oscillators are automatically controlled by the clock control system, see CLOCK — Clock control on page 54.

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 internal or external capacitors.

5.5.1 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 and is connected between pins **XC1** and **XC2**. For correct oscillation frequency, the load capacitance must match the specification in the crystal datasheet. The following figure shows how the 32 MHz crystal is connected to the high frequency crystal oscillator.

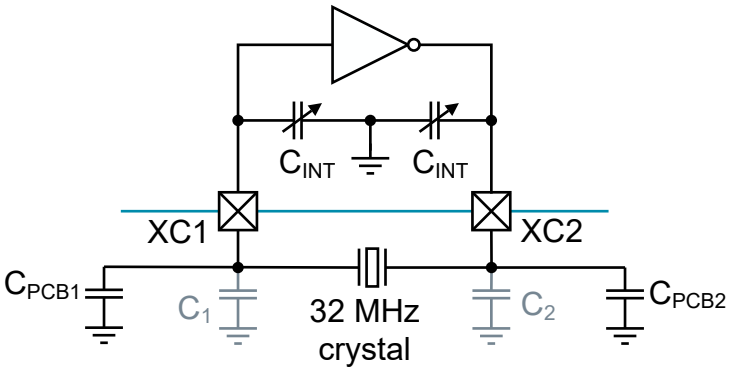


Figure 12: Circuit diagram of the high-frequency crystal oscillator

The device can be used with external capacitors C1 and C2 or the internal capacitors C_{INT}, which are configurable.

For reliable operation, the crystal load capacitance, shunt capacitance, equivalent series resistance, and drive level must comply with the specifications in table 32 MHz crystal oscillator (HFXO) on page 555. It is recommended to use a crystal with lower than maximum load capacitance and/or shunt capacitance. A low load capacitance reduces both start up time and current consumption.

When using internal capacitors, the load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is calculated by the following equation.

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

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

Figure 13: Load capacitance equation for internal capacitors

C_{INT} is the value of the internal capacitors. C_{pcb1} and C_{pcb2} are stray capacitance on the PCB.

The internal capacitor must be configured before starting the high-frequency crystal oscillator using the XOSTART task. To enable the internal capacitors, find the correct value for C_{INT} in the field OSCILLATORS.XOSC32M.CONFIG.INTCAP using the following equation.

```
INTCAP = (((CAPACITANCE-5.5)*(FICR->XOSC32MTRIM.SLOPE+791)) +
           FICR->XOSC32MTRIM.OFFSET*4)/256
```

The equation has the following variables:

- `CAPACITANCE` is the desired capacitor value of C_{INT} in pF, holding any value between 4.0 pF and 17.0 pF in 0.25 pF steps.
- `FICR->XOSC32MTRIM` are factory trim values which vary between devices.

After HFXO starts, the device uses the internal capacitor together with the external crystal after configuration.

5.5.1.1 Using external capacitors

It is possible to use external capacitors after disabling the internal capacitor.

When using external capacitors, 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 14: Load capacitance equation for external capacitors

C1 and C2 are the external capacitors. C_{pcb1} and C_{pcb2} are stray capacitance on the PCB. C_{pin} is the pin input capacitance on pins **XC1** and **XC2**.

When using external capacitors, disable the internal capacitor by setting OSCILLATORS.XOSC32M.CONFIG.INTCAP to `0`.

5.5.1.2 Crystal selection

Several crystals are supported by the 32 MHz crystal oscillator.

The following figure shows a simple model of a crystal. It has R-L-C series components, called equivalent series resistance (ESR), motional capacitance ($C_{0=N}$), and motional inductance (L_M). The capacitor in parallel, C_0 , is called the shunt capacitance, and models the package capacitance.

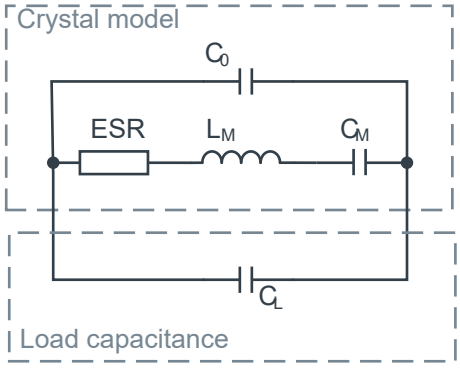


Figure 15: Simplified crystal model

The crystal needs to have parameters ESR, C_0 , and C_L selected to ensure the crystal oscillator is stable.

The following figure shows the maximum allowable combinations of ESR, C_0 and C_L for a given crystal. A crystal is supported if that crystal's parameters fall directly on the line or below it. Crystals that are above the line are not supported.

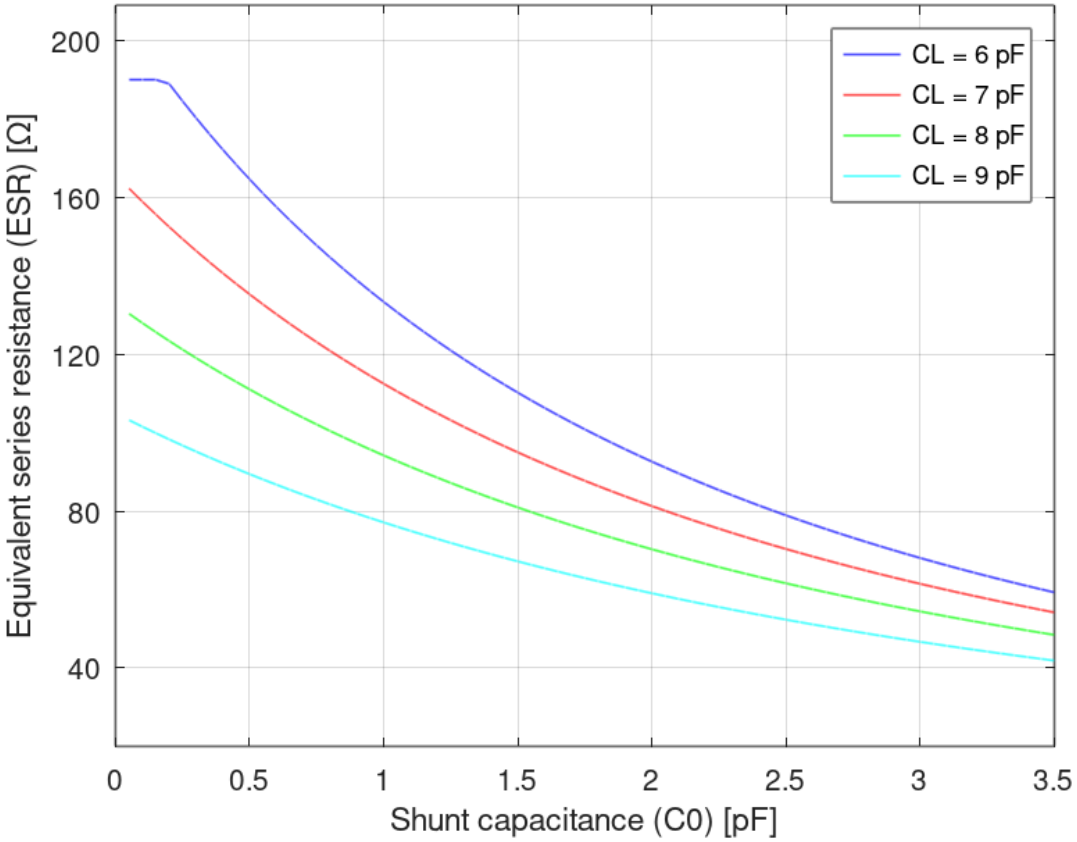


Figure 16: Maximum allowed combinations of ESR and C_0 for a given load capacitance C_L

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

For clock accuracy higher than LFRC, 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 shown in the following figure.

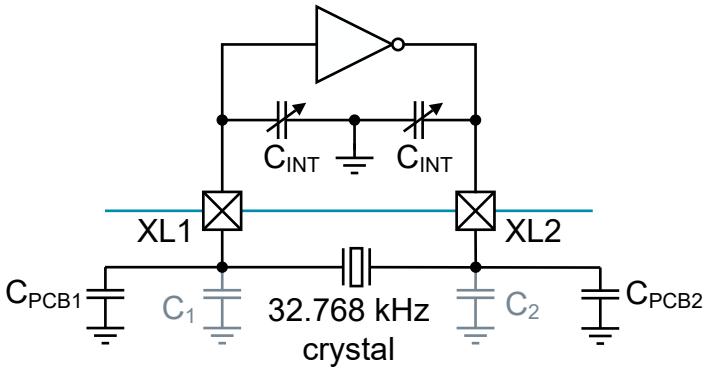


Figure 17: Circuit diagram of the low-frequency crystal oscillator

The device can be used with external capacitors C1 and C2 or the built-in configurable internal capacitors C_{INT}.

When using internal capacitors, the load capacitance (CL) is the total capacitance seen by the crystal across its terminals. It is calculated by the following equation.

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

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

Figure 18: Load capacitance equation for internal capacitors

C_{INT} is the value of the internal capacitors. C_{pcb1} and C_{pcb2} are stray capacitance on the PCB.

The internal capacitors must be configured before starting the low-frequency crystal oscillator (LFXO). To enable the internal capacitors, determine the correct field for OSCILLATORS.XOSC32KI.INTCAP using the following equation.

```
INTCAP = round( (2*CAPACITANCE - 12) * (FICR->XOSC32KTRIM.SLOPE + 0.765625 * 512)/512 +
 FICR->XOSC32KTRIM.OFFSET/64 )
```

The equation has the following variables:

- `CAPACITANCE` is the desired capacitor value in pF, holding any value between 3 pF and 18 pF in 0.65 pF steps.
- `FICR->XOSC32KTRIM` are factory trim values which are device specific.

When LFXO starts, it will use the internal capacitor together with the external crystal.

5.5.2.1 Using external capacitors

When using external capacitors, 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 19: Load capacitance equation for external capacitors

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. C_{pcb1} and C_{pcb2} are stray capacitance on the PCB. C_{pin} is the pin input capacitance on pins **XL1** and **XL2**. The load capacitors C1 and C2 must have the same value.

When using external capacitors, the internal capacitor is disabled by setting OSCILLATORS.XOSC32KI.INTCAP to `0`.

5.5.2.2 External source

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

The device can use a rail-to-rail clock, where the signal should be applied to the **XL1** pin with the **XL2** pin left unconnected or connected to GND. To enable rail-to-rail clock, set XOSC32KI.BYPASS=`Enabled`.

Using an external source requires that CLOCK.LFCLK.SRC=`LFXO`.

5.5.3 CPU clock frequency selection

The CPU clock frequency is configurable on boot in the register PLL.FREQ (Retained) on page 77.

The device supports 64 or 128 MHz frequency.

The device starts at 64 MHz. For higher frequencies, it must be configured when the CPU starts and before any peripherals that use the high-frequency clock are enabled. Changing the frequency on a running system or to an unsupported value causes undefined system behavior and the device can malfunction.

5.5.4 Registers

Instances

Instance	Domain	Base address	Description
OSCILLATORS	GLOBAL	0x40120000	Oscillator control

Register overview

Register	Offset	Description
XOSC32M.CONFIG.INTCAP	0x71C	Crystal load capacitor as seen by the crystal across its terminals, including pin capacitance but excluding PCB stray capacitance.
PLL.FREQ	0x800	Set speed of MCU power domain, including CPU This register is retained.
PLL.CURRENTFREQ	0x804	Current speed of MCU power domain, including CPU This register is retained.
XOSC32KI.BYPASS	0x900	Enable or disable bypass of LFCLK crystal oscillator with external clock source
XOSC32KI.INTCAP	0x904	Programmable capacitance of XL1 and XL2 This register is retained.

5.5.4.1 XOSC32M

32 MHz oscillator control

5.5.4.1.1 XOSC32M.CONFIG.INTCAP

Address offset: 0x71C

Crystal load capacitor as seen by the crystal across its terminals, including pin capacitance but excluding PCB stray capacitance.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																											A	A	A	A	A	A
Reset 0x00000020	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	VAL			Crystal load capacitor value Use the provided equation in OSCILLATORS — Oscillator control on page 73 to calculate the register value.

5.5.4.2 PLL

Oscillator control

5.5.4.2.1 PLL.FREQ (Retained)

Address offset: 0x800

Set speed of MCU power domain, including CPU

This register is retained.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000003				**0 0 0 0 0 0 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**
ID	R/W	Field	Value ID	Value	Description
A	RW	FREQ			Select CPU speed
			CK128M	1	128 MHz
			CK64M	3	64 MHz

5.5.4.2.2 PLL.CURRENTFREQ (Retained)

Address offset: 0x804

Current speed of MCU power domain, including CPU

This register is retained.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000003				**0 0 0 0 0 0 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**
ID	R/W	Field	Value ID	Value	Description
A	R	CURRENTFREQ			Active CPU speed
			CK128M	1	128 MHz
			CK64M	3	64 MHz

5.5.4.3 XOSC32KI

32.768 kHz oscillator control

5.5.4.3.1 XOSC32KI.BYPASS

Address offset: 0x900

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 crystal)
			Enabled	1	Enable (use rail-to-rail external source)

5.5.4.3.2 XOSC32KI.INTCAP (Retained)

Address offset: 0x904

Programmable capacitance of XL1 and XL2

Use the provided equation in OSCILLATORS — Oscillator control on page 73 to calculate the register value.

This register is retained.

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A
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	VAL			Crystal load capacitor as seen by the crystal across its terminals, including pin capacitance but excluding PCB stray capacitance. Use the provided equation in "Using internal capacitors section" to calculate the register value.

5.6 POWER — Power control

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

The POWER peripheral requests resources from the power and clock subsystem. The power and clock subsystem makes sure that the power mode with the proper latency settings is selected when requested. This means that the Constant Latency mode is prioritized over Low-power mode. For an overview of power modes, see Sub-power modes on page 51.

The event POFWARN is a system level event that enables the device to react quickly if there is a power failure. The power-fail comparator must be configured and enabled to receive the event, see Power-fail comparator on page 53 for more information.

Power control of the RAM blocks is controlled by the memory configuration peripheral (MEMCONF), see MEMCONF — Memory configuration on page 30.

Note: Registers INTEN on page 82, INTENSET on page 83, and INTENCLR on page 83 are shared between the POWER and CLOCK peripherals.

5.6.1 Registers

Instances

Instance	Domain	Base address	Description
POWER	GLOBAL	0x4010E000	Power control

Register overview

Register	Offset	Description
TASKS_CONSTLAT	0x30	Enable Constant Latency mode
TASKS_LOWPWR	0x34	Enable Low-power mode (variable latency)
SUBSCRIBE_CONSTLAT	0xB0	Subscribe configuration for task CONSTLAT
SUBSCRIBE_LOWPWR	0xB4	Subscribe configuration for task LOWPWR
EVENTS_POFWARN	0x130	Power failure warning
EVENTS_SLEEPENTER	0x134	CPU entered WFI/WFE sleep
EVENTS_SLEEPEXIT	0x138	CPU exited WFI/WFE sleep
PUBLISH_POFWARN	0x1B0	Publish configuration for event POFWARN
PUBLISH_SLEEPENTER	0x1B4	Publish configuration for event SLEEPENTER
PUBLISH_SLEEPEXIT	0x1B8	Publish configuration for event SLEEPEXIT
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt

Register	Offset	Description
GPREGRET[n]	0x500	General purpose retention register This register is retained.
CONSTLATSTAT	0x520	Status of constant latency

5.6.1.1 TASKS_CONSTLAT

Address offset: 0x30

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

5.6.1.2 TASKS_LOWPWR

Address offset: 0x34

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

5.6.1.3 SUBSCRIBE_CONSTLAT

Address offset: 0xB0

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																									A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CONSTLAT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.6.1.4 SUBSCRIBE_LOWPWR

Address offset: 0xB4

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task LOWPWR will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

5.6.1.5 EVENTS_POFWARN

Address offset: 0x130

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

5.6.1.6 EVENTS_SLEEPENTER

Address offset: 0x134

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

5.6.1.7 EVENTS_SLEEPEXIT

Address offset: 0x138

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

5.6.1.8 PUBLISH_POFWARN

Address offset: 0x1B0

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		[0..255]	DPPI channel that event POFWARN will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.6.1.9 PUBLISH_SLEEPENTER

Address offset: 0x1B4

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		[0..255]	DPPI channel that event SLEEPENTER will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.6.1.10 PUBLISH_SLEEPEXIT

Address offset: 0x1B8

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		[0..255]	DPPI channel that event SLEEPEXIT will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

5.6.1.11 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

5.6.1.12 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 W1S	POFWARN			Write '1' to enable interrupt for event POFWARN
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	SLEEPENTER			Write '1' to enable interrupt for event SLEEPENTER
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	SLEEPEXIT			Write '1' to enable interrupt for event SLEEPEXIT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

5.6.1.13 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 W1C	POFWARN			Write '1' to disable interrupt for event POFWARN
			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
B	RW W1C	SLEEPENTER			Write '1' to disable interrupt for event SLEEPENTER
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	SLEEPEXIT			Write '1' to disable interrupt for event SLEEPEXIT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

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

Address offset: 0x500 + (n × 0x4)

General purpose retention register

This register is retained.

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

5.6.1.15 CONSTLATSTAT

Address offset: 0x520

Status of constant 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	R	STATUS			Status
			Disable	0	Constant latency disabled
			Enable	1	Constant latency enabled

5.7 REGULATORS — Regulator control

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

All system components are powered from the main on-chip voltage regulator, VREGMAIN. The regulator converts the voltage supplied on **VDD** to internal voltage.

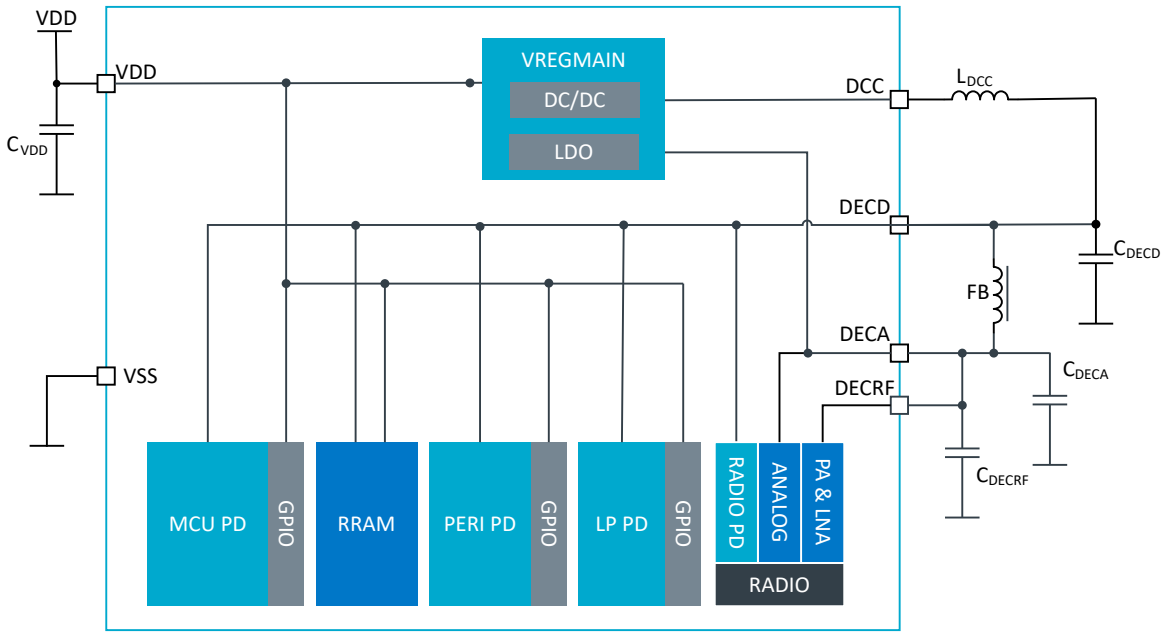


Figure 20: Regulator configuration

The main supply voltage is connected to the **VDD** pin.

5.7.1 VREGMAIN — Main regulator

VREGMAIN is the main regulator of the system.

The device only supports DC/DC mode, which needs external components. For details, see Reference circuitry on page 552.

VREGMAIN starts up using a fall-back mode, and the DC/DC regulator must be enabled using register VREGMAIN.DCDCEN on page 87.

Inductor detection

The DC/DC regulator requires an external inductor connected to the **DCC** pin. If an inductor is not present, the regulator stays in fall-back mode to enable a soft shut-down of the system. Register VREGMAIN.INDUCTORDET on page 87 reports the inductor detection status and can be used to detect inductor failures that prevent DC/DC operation.

5.7.2 Registers

Instances

Instance	Domain	Base address	Description
REGULATORS	GLOBAL	0x40120000	Regulator control

Register overview

Register	Offset	Description
SYSTEMOFF	0x500	System OFF register
POFCON	0x530	Power-fail comparator configuration This register is retained.
POFSTAT	0x534	Power-fail comparator status register
VREGMAIN.DCDCEN	0x600	Enable DC/DC converter
VREGMAIN.INDUCTORDET	0x604	VREGMAIN inductor detection

5.7.2.1 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

5.7.2.2 POFCON (Retained)

Address offset: 0x530

Power-fail comparator configuration

This register is retained.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									C			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
			V17	0	Set threshold to 1.7 V for VDD
			V18	1	Set threshold to 1.8 V for VDD
			V19	2	Set threshold to 1.9 V for VDD
			V20	3	Set threshold to 2.0 V for VDD
			V21	4	Set threshold to 2.1 V for VDD
			V22	5	Set threshold to 2.2 V for VDD
			V23	6	Set threshold to 2.3 V for VDD
			V24	7	Set threshold to 2.4 V for VDD
			V25	8	Set threshold to 2.5 V for VDD
			V26	9	Set threshold to 2.6 V for VDD
			V27	10	Set threshold to 2.7 V for VDD
			V28	11	Set threshold to 2.8 V for VDD
			V29	12	Set threshold to 2.9 V for VDD
			V30	13	Set threshold to 3.0 V for VDD
			V31	14	Set threshold to 3.1 V for VDD

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

ID	R/W	Field	Value ID	Value	Description
			V32	15	Set threshold to 3.2 V for VDD
C	RW	EVENTDISABLE			Disable the POFWARN power-fail warning event
			Enabled	0	POFWARN event is generated
			Disabled	1	POFWARN event is not generated

5.7.2.3 POFSTAT

Address offset: 0x534

Power-fail comparator 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	COMPARATOR			Power-fail comparator status
			Above	0	Voltage detected above VPOF threshold
			Below	1	Voltage detected below VPOF threshold

5.7.2.4 VREGMAIN

Register interface for main voltage regulator.

5.7.2.4.1 VREGMAIN.DCDCEN

Address offset: 0x600

Enable DC/DC converter

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

ID	R/W	Field	Value ID	Value	Description
A	RW	VAL			Enable DC/DC buck converter
			Disabled	0	Disable DC/DC buck converter
			Enabled	1	Enable DC/DC converter If inductor is not present (see register VREGMAIN.INDUCTORDET), the DC/DC converter cannot operate

5.7.2.4.2 VREGMAIN.INDUCTORDET

Address offset: 0x604

VREGMAIN inductor detection

Ensure that an inductor is connected to the DCC pin. The detection can only take place before the DC/DC converter is enabled.

Note: The DC/DC converter cannot operate without an inductor.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	DETECTED			
			InductorNotDetected	0	VREGMAIN inductor not detected
			InductorDetected	1	VREGMAIN inductor detected

5.8 RESET — Reset control

A system-level reset is triggered by the following resets:

- Brownout
- Power-on
- CTRL-AP
- Watchdog
- Wakeup from System OFF
- Tamper detection
- CPU Lockup
- Pin

The system reset sources are shown in the following figure.

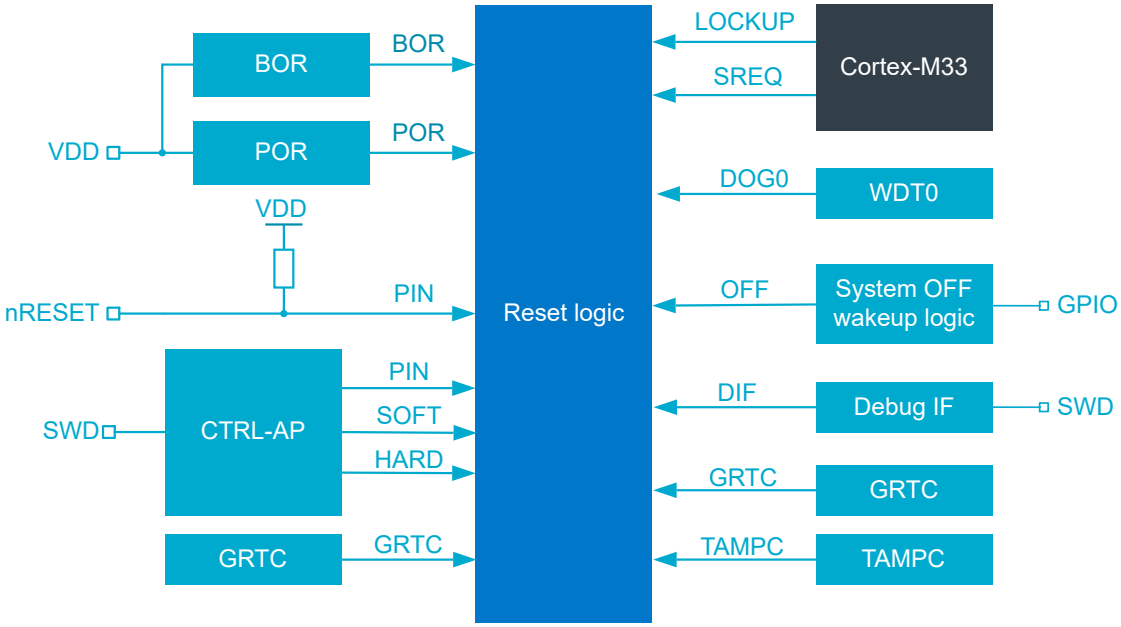


Figure 21: Reset sources

After a reset, the device automatically starts up. The register RESETREAS on page 91 can be read to determine which source generated the reset.

5.8.1 Power-on reset

The power-on reset (POR) generator initializes the system when the supply voltage is above the power-on threshold.

The system is held in a reset state until the supply reaches the minimum operating voltage and the internal voltage regulators start. After a power-on reset, the device starts up.

5.8.2 Pin reset

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

Pin reset is available on the reset pin **nRESET**, see Pin assignments on page 546. The device starts after **nRESET** is deasserted.

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 182.

CTRL-AP can also reset a pin, see CTRL-AP resets on page 89.

5.8.3 Brownout reset

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

Similar to a power-on reset, the device starts after BOR is deasserted.

5.8.4 Wakeup from System OFF mode reset

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

Similar to a power-on reset (POR), the device is started after waking up from System OFF.

If the device is in Debug interface mode, the debug access port (DAP) is not reset after a wakeup from System OFF mode. For more information, see Debug on page 526.

For details on the System OFF mode, see System OFF mode on page 52.

5.8.5 Soft reset

A soft reset is generated when the SYSRESETREQ bit of the application interrupt and reset control register (AIRCR) in the Arm CPU is set. For more information, see Arm documentation.

A soft reset can also be generated using CTRL-AP, see CTRL-AP resets on page 89.

Similar to a power-on reset (POR), the device is restarted after a soft reset.

5.8.6 CTRL-AP resets

CTRL-AP can generate the following resets.

- Soft reset
- Pin reset
- Hard reset. This is used during an Erase ALL operation and is less intrusive than Pin reset. For more details, see Reset behavior on page 90.

Through the debugger interface, CTRL-AP can generate three resets using register RESET on page 536. For more details, see CTRL-AP — Control access port on page 531.

Similar to a power-on reset (POR), the device is restarted after a CTRL-AP reset.

5.8.7 Watchdog timer reset

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

Similar to a power-on reset (POR), the device is started after a watchdog reset.

5.8.8 Retained registers

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

5.8.9 Reset behavior

The reset source determines the behavior of the device after a reset.

In System OFF mode, the watchdog timer is not running and CPU lockup is not possible. RAM may be fully or partially retained, depending on RAM retention settings in MEMCONF — Memory configuration on page 30.

If the device is in Debug Interface mode, the debug components are not reset. Additionally, CPU lockup does not generate a reset. See Debug on page 526 for more information about the different debug components in the system.

An 'x' in the table means that the specific module or register is reset. The table also explicitly lists which reset sources are commonly referred to as 'cold boot'.

							Retained registers			
Reset source	**Cold boot**	**CM33**	**Peripherals**	**Debug**	**RAM**	**WDT**	**REGULATORS and OSCILLATORS and CPU speed**	**RESETREAS**	**POWERGPRETREG**	**GPIO**	**GRTC.SYSCOUNTER**
CPU lockup		x	x							x[1]	x
Soft reset and CTRL-AP soft reset		x	x							x[1]	
Wakeup from System Off mode		x	x			x					
CTRL-AP hard reset		x	x	x	x	x	x			x	x
CTRL-AP pin reset		x	x	x	x	x	x			x	x
Watchdog timer reset		x	x	x	x	x	x			x	x
Pin reset		x	x	x	x	x	x			x	x
TAMPC reset		x	x	x	x	x	x			x	x
Brownout reset	x	x	x	x	x	x	x	x	x	x	x
Power-on reset	x	x	x	x	x	x	x	x	x	x	x

Table 20: Reset overview

[1] Except the CTRLSEL field.

For TAMPC reset sources, see TAMPC — Tamper controller on page 123.

5.8.10 Registers

Instances

Instance	Domain	Base address	Description
RESET	GLOBAL	0x4010E000	Reset status

Register overview

Register	Offset	Description
RESETREAS	0x600	Reset reason

5.8.10.1 RESETREAS

Address offset: 0x600

Reset reason

Before entering System OFF mode, the RESETREAS register must be cleared.

Note: Unless cleared, the RESETREAS register will be cumulative. A field is cleared by writing `1` to it.

Note: When RESETREAS shows a pin reset (RESETPIN), ignore other reset reason 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																			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
					CTRL-AP generating a pin reset has its own bit.
			NotDetected	0	Not detected
			Detected	1	Detected
B	RW	DOG0			Reset from watchdog timer 0 detected
			NotDetected	0	Not detected
			Detected	1	Detected
C	RW	CTRLAPSOFT			Soft reset from CTRL-AP detected
			NotDetected	0	Not detected
			Detected	1	Detected
D	RW	CTRLAPHARD			Reset due to CTRL-AP hard reset
			NotDetected	0	Not detected
			Detected	1	Detected
E	RW	CTRLAPPIN			Reset due to CTRL-AP pin reset
			NotDetected	0	Not detected
			Detected	1	Detected
F	RW	SREQ			Reset from soft reset detected
			NotDetected	0	Not detected
			Detected	1	Detected
G	RW	LOCKUP			Reset from CPU lockup detected
			NotDetected	0	Not detected
			Detected	1	Detected
H	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
I	RW	DIF			Reset triggered by Debug Interface
			NotDetected	0	Not detected
			Detected	1	Detected
J	RW	GRTC			Reset due to wakeup from GRTC
			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																						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																															
			Detected	1																															
K	RW	SECTAMPER																																	
			NotDetected	0																															
			Detected	1																															

6 Event system

The distributed programmable peripheral interconnect (DPPI) system enables peripherals to interact autonomously with each other through tasks and events, without intervention from the CPU.

The DPPI channels are local to each power domain, but can be transferred between power domains using PPI bridges.

The following figure shows the power domains, the PPI controllers (DPPIC), and the PPI bridges (PPIB).

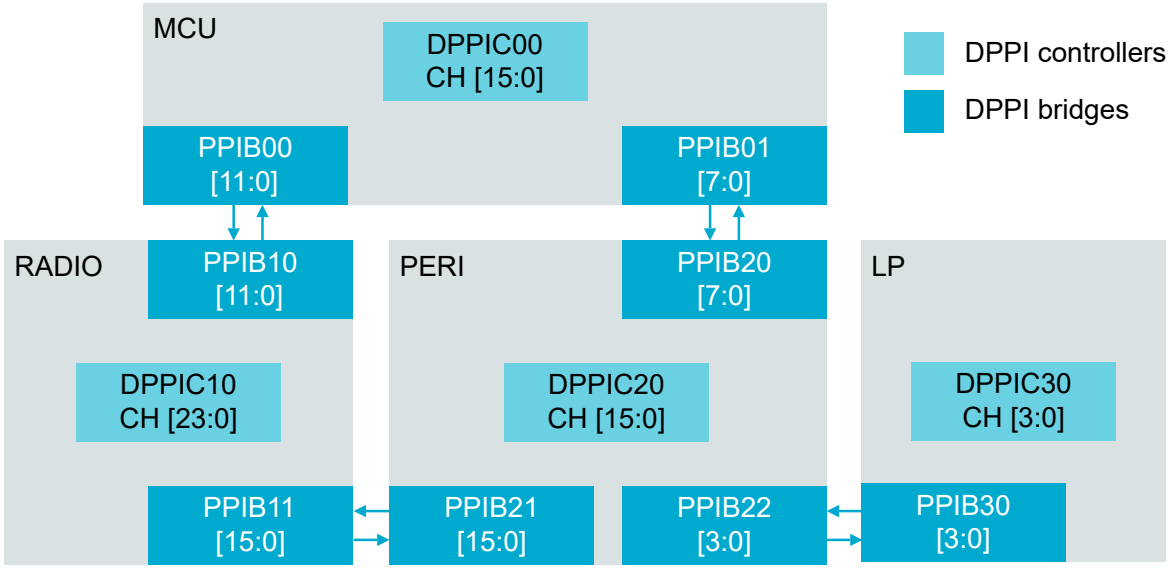


Figure 22: Power domains and PPI bridges

A subset of PPI channels from a power domain can be bridged across to a different power domain. For example, PPIB00 can bridge a set number of configurable DPPI channels to PPIB10. For more details on how to configure the PPI and bridge system, see DPPI — Distributed programmable peripheral interconnect on page 94 and PPIB — PPI Bridge on page 102.

6.1 DPPI latencies

DPPI task and event latency depends on the power domain of the source and destination peripherals.

DPPI signals operate on the HCLKCORE, PCLK32M, and PCLK16M clocks.

Power domain	Clock source
MCU	HCLKCORE (64 or 128 MHz)
RADIO	PCLK32M (32 MHz)
PERI	PCLK16M (16 MHz)
LP	PCLK16M (16 MHz)

Table 21: DPPI clock frequency

For peripherals in the same power domain, there is a two cycle delay from when an event is generated until a task subscribing to the same channel is triggered. Events that are generated while the system

is sleeping will have additional access latency, as the system needs to request and provide sufficient resources for PPI handling. Examples of such events are the following:

- A GPIO toggling and generating an event through GPIOTE
- Events generated by GRTC while all other clocks are stopped

To improve PPI latency, the Constant Latency mode can be used, see Sub-power modes on page 51.

For peripherals in different power domains, additional access latency will apply. Events that are generated while the generating or receiving power domains are sleeping will have additional access latency, as the system needs to request and provide sufficient resources for PPI handling.

6.2 DPPI — Distributed programmable peripheral interconnect

The distributed programmable peripheral interconnect (DPPI) enables peripherals to interact autonomously with each other through tasks and events, without CPU intervention. 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 138.

The main features of DPPI are the following:

- 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. These modules are connected to a fixed number of DPPI channels and a DPPI controller (DPPIC).

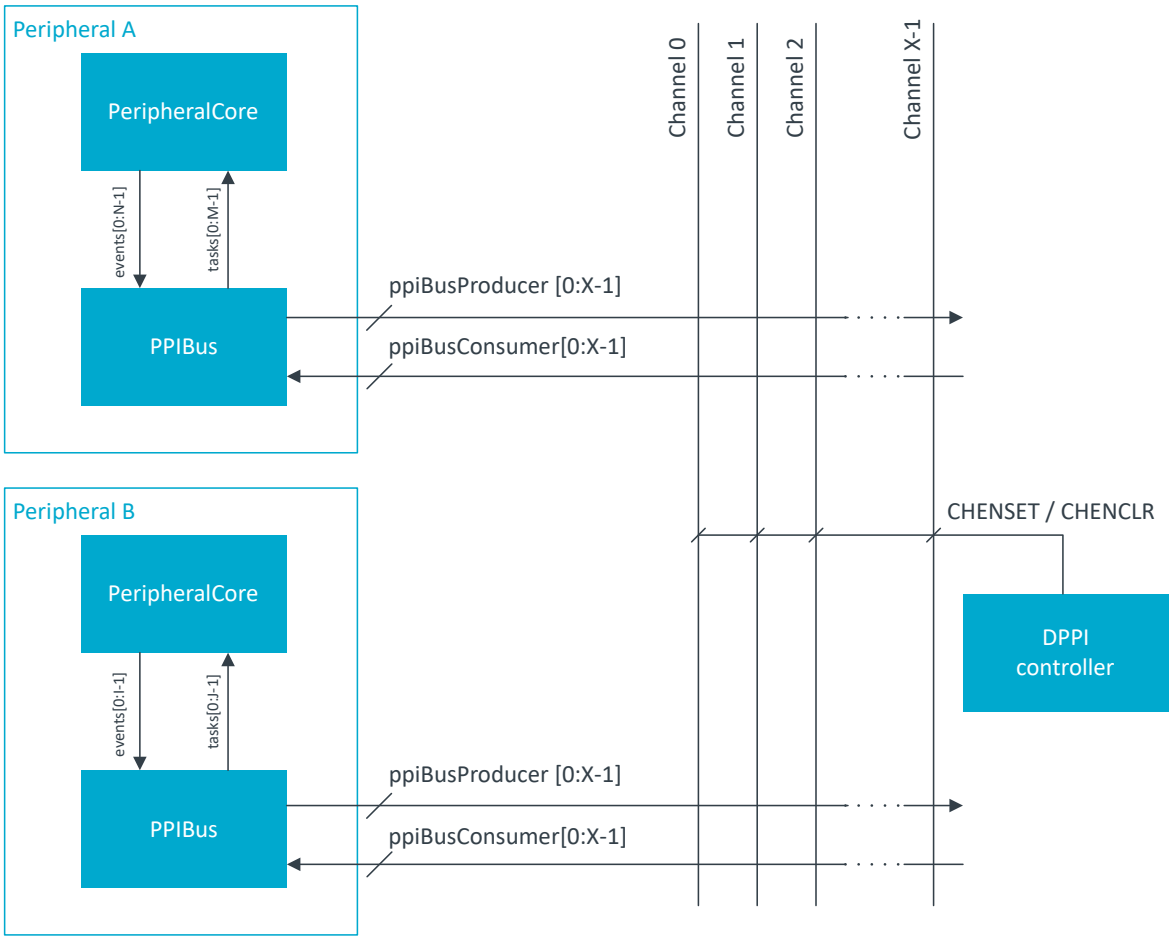


Figure 23: DPPI overview

6.2.1 Channel publish and subscribe

The DPPI system directs peripheral events to a channel (publishing) and converts events from a channel into peripheral tasks (subscribing).

All peripherals include the following:

- One publish register per event
- One subscribe register per task

The publish and subscribe registers have two fields. A channel index field (CHIDX) determines which channel an event is published to, or which channel a task is subscribed to. In addition there is an ENABLE field that must be set to `1` to activate the operation.

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.

The following figure shows how peripheral events are routed onto different channels based on publish registers.

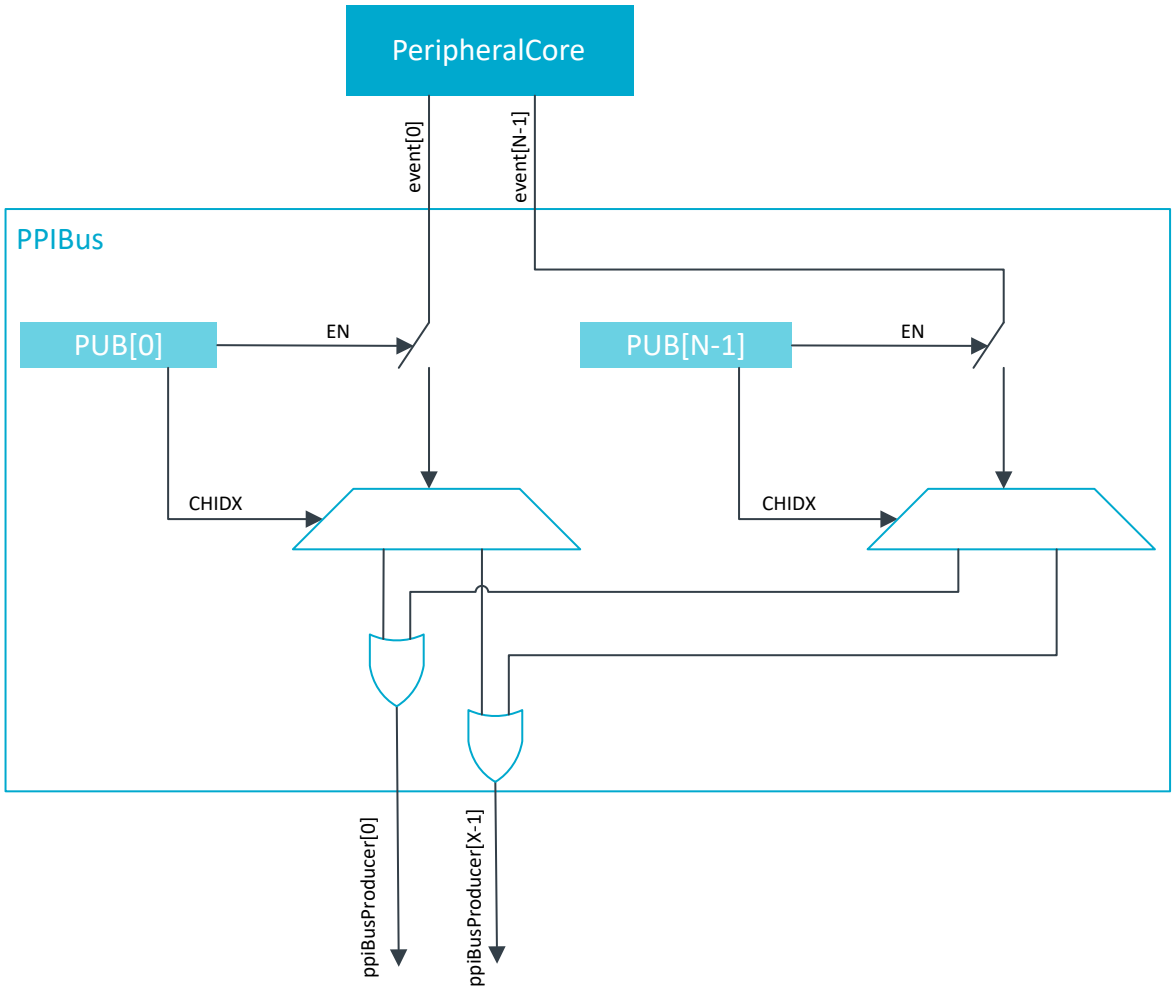


Figure 24: DPPI events flow

The following figure illustrates how peripheral tasks are triggered from different channels based on subscribe registers.

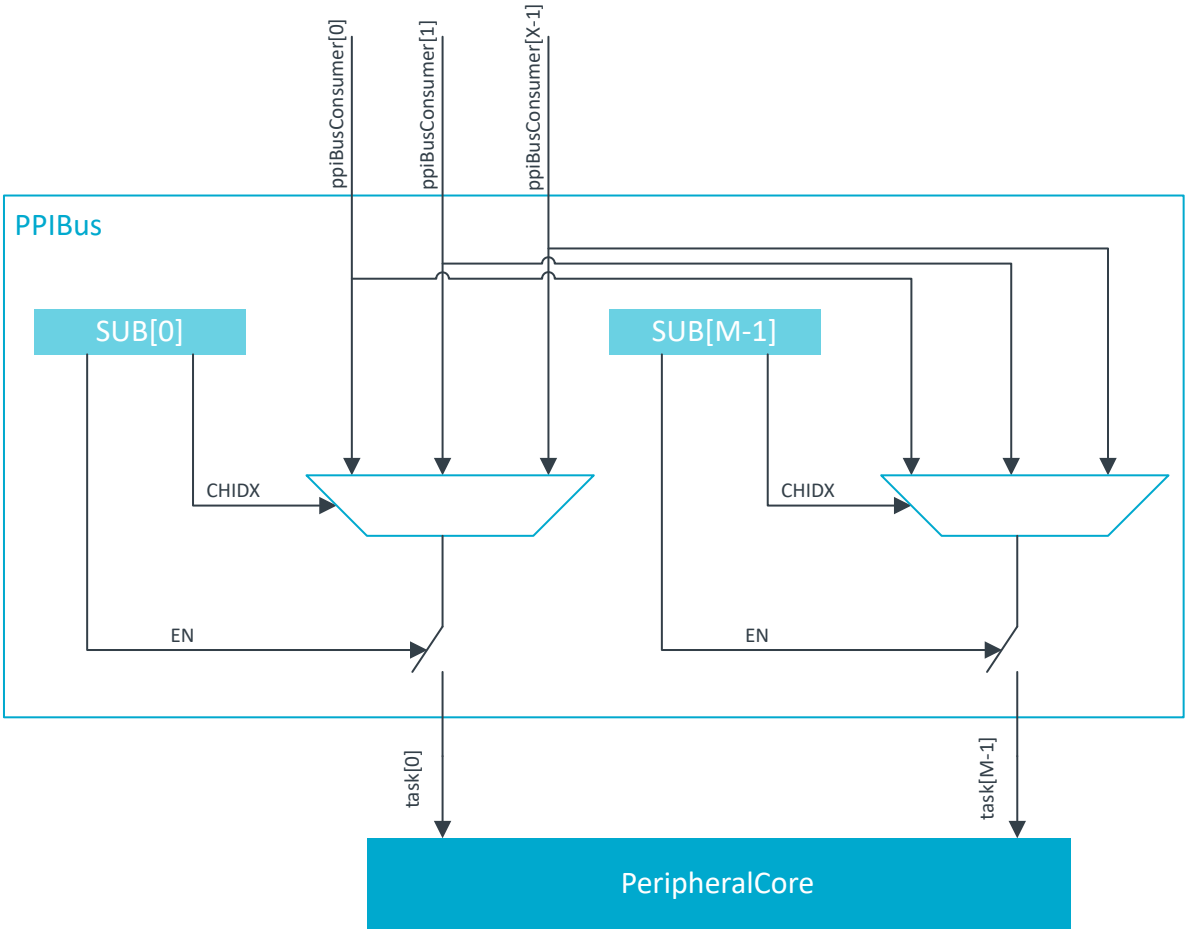


Figure 25: DPPI tasks flow

6.2.2 DPPI controller (DPPIC)

Enabling and disabling of DPPI channels is handled through DPPIC.

There are two ways of enabling or disabling a DPPI channel using DPPIC:

- 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. Channel groups should be defined via the CHG registers before these tasks are triggered.

Note: ENABLE tasks are prioritized over DISABLE tasks, i.e. in case of a simultaneously occurring TASKS_CHG[m].EN and TASKS_CHG[n].DIS (m and n can be equal or different), the CHG[m].EN task will be prioritized if the same channel subscribed to both groups.

DPPIC 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[n], the corresponding CHG[n].EN and CHG[n].DIS subscribe registers must be disabled. Writes to CHG[n] are ignored if any of the two subscribe registers are enabled.

6.2.3 Connection examples

Several connection options are available with DPPI. 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 the TIMER compare register and the SAADC start task.

The channel configuration is set up first. TIMER will publish its COMPARE0 event on channel 0, and SAADC will subscribe its START task to events on the same channel. When this is finished, the channel is enabled through the DPPIC.

```
NRF_TIMER20->PUBLISH_COMPARE[0] = (0 << TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
 TIMER_PUBLISH_COMPARE_EN_Msk;
NRF_SAADC->SUBSCRIBE_START = (0 << SAADC_SUBSCRIBE_START_CHIDX_Pos) |
 SAADC_SUBSCRIBE_START_EN_Msk;
NRF_DPPIC20->CHENSET = DPPIC_CHENSET_CH0_Msk;
```

Many-to-many connection

The following 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 TIMER 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_DPPIC20->CHG[0] = (DPPIC_CHG_CH0_Included << DPPIC_CHG_CH0_Pos);
NRF_GPIOTE20->PUBLISH_IN[0] = (0 << GPIOTE_PUBLISH_IN_CHIDX_Pos) |
 GPIOTE_PUBLISH_IN_EN_Msk;
NRF_TIMER20->PUBLISH_COMPARE[0] = (0 << TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
 TIMER_PUBLISH_COMPARE_EN_Msk;
NRF_SAADC->SUBSCRIBE_START = (0 << SAADC_SUBSCRIBE_START_CHIDX_Pos) |
 SAADC_SUBSCRIBE_START_EN_Msk;
NRF_DPPIC20->SUBSCRIBE_CHG[0].DIS = DPPIC_CHENSET_CH0_Msk | DPPIC_SUBSCRIBE_CHG_DIS_EN_Msk;
NRF_DPPIC20->TASK_CHG[0].EN = 1;
```

6.2.4 Registers

Instances

Instance	Domain	Base address	Description
DPPIC00	GLOBAL	0x40042000	DPPI controller DPPIC00
DPPIC10	GLOBAL	0x40082000	DPPI controller DPPIC10
DPPIC20	GLOBAL	0x400C2000	DPPI controller DPPIC20
DPPIC30	GLOBAL	0x40102000	DPPI controller DPPIC30

Configuration

Instance	Domain	Configuration
DPPIC00	GLOBAL	16 DPPI channels 2 DPPI groups
DPPIC10	GLOBAL	24 DPPI channels 6 DPPI groups
DPPIC20	GLOBAL	16 DPPI channels 6 DPPI groups
DPPIC30	GLOBAL	4 DPPI channels 2 DPPI groups

Register overview

Register	Offset	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

6.2.4.1 TASKS_CHG[n] (n=0..5)

Channel group tasks

6.2.4.1.1 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

6.2.4.1.2 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**
ID	R/W	Field	Value ID	Value	Description
A	W	DIS			Disable channel group n
			Trigger	1	Trigger task

6.2.4.2 SUBSCRIBE_CHG[n] (n=0..5)

Subscribe configuration for tasks

6.2.4.2.1 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CHG[n].EN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.2.4.2.2 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**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CHG[n].DIS will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.2.4.3 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				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-X	RW	CH[i] (i=0..23)			Enable or disable channel i
			Disabled	0	Disable channel
			Enabled	1	Enable channel

6.2.4.4 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												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-X	RW W1S	CH[i] (i=0..23)										Channel i enable set register. Writing 0 has no effect.																						
			Disabled	0								Read: Channel disabled																						
			Enabled	1								Read: Channel enabled																						
			Set	1								Write: Enable channel																						

6.2.4.5 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												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-X	RW W1C	CH[i] (i=0..23)										Channel i enable clear register. Writing 0 has no effect.																						
			Disabled	0								Read: Channel disabled																						
			Enabled	1								Read: Channel enabled																						
			Clear	1								Write: Disable channel																						

6.2.4.6 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												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-X	RW	CH[i] (i=0..23)										Include or exclude channel i																						
			Excluded	0								Exclude																						
			Included	1								Include																						

6.3 PPIB — PPI Bridge

PPIB connects tasks and events of peripherals in two different PPI systems in different power-domains.

A PPI system contains a number of peripherals that can communicate with each other by using tasks and events. This functionality is enabled by the DPPI peripheral. In a PPI system, the peripherals and DPPI are instantiated in the same APB bus.

The following figure shows a PPI system including a PPIB:

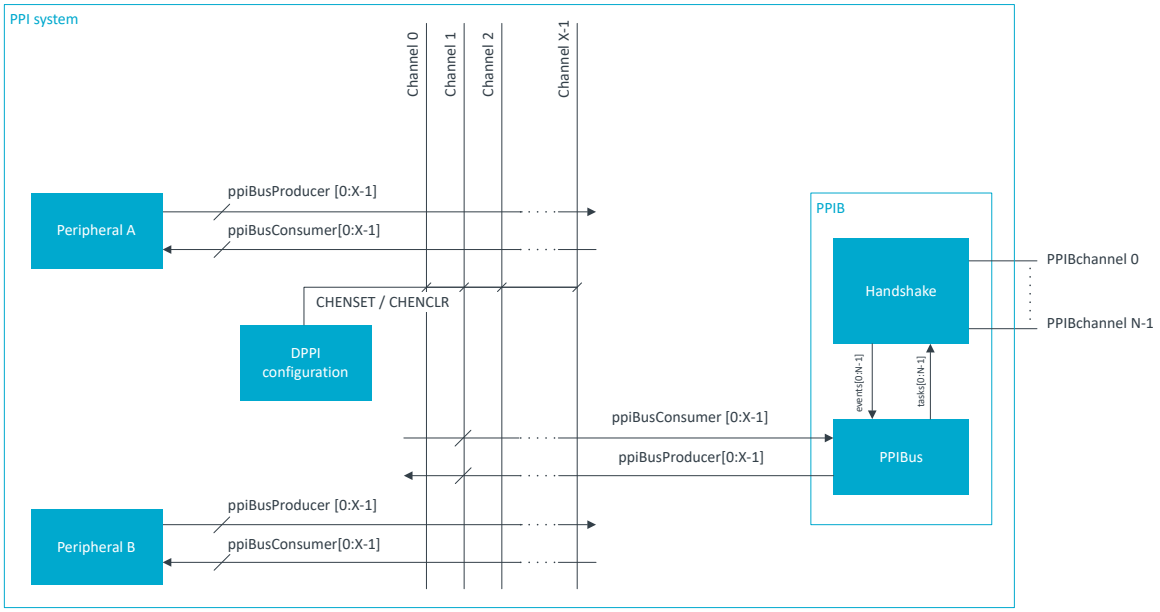


Figure 26: PPI system with PPIB

PPIB uses tasks and events like a standard peripheral, and connects to local DPPI channels via PPIBus. For more information on PPIBus module, see DPPI.

PPIB has a number of channels. Each PPIB channel connects to a single DPPI channel.

6.3.1 PPIB connections

A PPIB channel in one PPI system can be connected to a PPIB channel in another PPI system forming a PPIB connection.

A channel belonging to a PPIB instance in a PPI system is connected to a channel belonging to a PPIB instance in a different PPI system, creating a one-to-one PPIB connection between the two PPI systems. The connections are fixed and point-to-point, that is, a channel in a PPIB instance is connected only to a specific channel in another PPIB instance. For information on how the channels in the different PPIB instances are connected, see the Configuration table under the Registers section.

A PPIB channel can be configured as either source or sink. When configuring one side of the PPIB connection as source, the other side of the PPIB connection must be configured as sink, and viceversa. PPIB connections are unidirectional. Configuring both sides of a connection as source and sink at the same time will yield unexpected results.

On the source side of a PPIB connection, in order to send a (local) peripheral event to a different PPI system, the corresponding PPIB channel is configured as a consumer, subscribing to the same DPPI channel as the (local) peripheral publishes to, using the PPIB.SUBSCRIBE_SEND[n] register, with n the PPIB channel number.

On the sink side of a PPIB connection, for a (local) peripheral to be able to receive this event, the corresponding PPIB channel is configured as a producer, publishing to the same DPPI channel as the (local) peripheral subscribes to, using the PPIB.PUBLISH_RECEIVE[n] register, with n the PPIB channel number.

In a PPI system, several peripherals can publish to the same DPPI channel on the source side of a PPIB connection. Similarly, several peripherals can subscribe to the same DPPI channel on the sink side of a PPIB connection. This allows multiple connection options between peripherals in different PPI systems, same as DPPI allows in a local PPI system: one-to-one, one-to-many, many-to-one and many-to-many. However, when multiple peripherals can publish to the same DPPI channel on the source side of a PPIB connection, there is a risk of overflow. See Handshake and overflow on page 103.

6.3.2 Handshake and overflow

The two PPIB instances in a PPIB connection need a handshake to transfer a peripheral event.

This is handled by a Handshake module in the PPIB. If a handshake fails because an earlier event has not been processed completely, the new event won't be sent. Instead, bit `i` in OVERFLOW.SEND register on the source side will be set, with `i` the correspoding PPIB channel number.

6.3.3 Connection examples

This section contains examples on how two connect two PPI systems using PPIB.

The following example shows how to create a PPIB connection between the TIMER10 compare event in the RADIO PD and the SAADC start task in PERI PD. PPIB11 in RADIO PD is hardwired to PPIB21 in PERI PD, which allows the PPI systems in the two separate power domains to connect. DPPI channel 0 is used by both power domains. Note that it is only necessary to use the same DPPI channel within the power domain; different DPPI channels can be used across power domains. An example of this is given further down in this section.

```
   // RADIO PD
   NRF_TIMER10->PUBLISH_COMPARE[0] = (0<<TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
TIMER_PUBLISH_COMPARE_EN_Msk;
   NRF_PPIB11->SUBSCRIBE_SEND[0] = (0<<PPIB_SUBSCRIBE_SEND_CHIDX_Pos) |
PPIB_SUBSCRIBE_SEND_EN_Msk;
   NRF_DPPIC10->CHENSET = DPPIC_CHENSET_CH0_Msk;

   // PERI PD
   NRF_SAADC->SUBSCRIBE_START = (0<<SAADC_SUBSCRIBE_START_CHIDX_Pos) |
SAADC_SUBSCRIBE_START_EN_Msk;
   NRF_PPIB21->PUBLISH_RECEIVE[0] = (0<<PPIB_PUBLISH_RECEIVE_CHIDX_Pos) |
PPIB_PUBLISH_RECEIVE_EN_Msk;
   NRF_DPPIC20->CHENSET = DPPIC_CHENSET_CH0_Msk;
```

The following example shows how to create a PPIB connection between the TIMER10 compare event in the RADIO PD and the GPIOTE SET task in LP PD. The two PPI systems must be connected through PPIB instances PPIB21 and PPIB22 in PERI PD. These PPIB instances are not connected to any peripheral, only to the PPIB instances in RADIO and LP power domains. PERI PD acts as a central system that connects the two systems by means of local PPIB and DPPIC instances. This allows scaling to larger PPI systems, since multiple PPI systems can be interconnected through a central PPI system. DPPI channel 0 is used for internal RADIO PD connections, channel 1 is used by LP PD, and channel 5 is used by PERI PD. It is

important that same DPPI channels are used within a power domain, but across domains the DPPI channel number does not matter.

```
   // RADIO PD
   NRF_TIMER10->PUBLISH_COMPARE[0] = (0<<TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
TIMER_PUBLISH_COMPARE_EN_Msk;
   NRF_PPIB11->SUBSCRIBE_SEND[0] = (0<<PPIB_SUBSCRIBE_SEND_CHIDX_Pos) |
PPIB_SUBSCRIBE_SEND_EN_Msk;
   NRF_DPPIC10->CHENSET = DPPIC_CHENSET_CH0_Msk;

   // LP PD
   NRF_GPIOTE30->SUBSCRIBE_START = (1<<GPIOTE_SUBSCRIBE_SET_CHIDX_Pos) |
GPIOTE_SUBSCRIBE_SET_EN_Msk;
   NRF_PPIB30->PUBLISH_RECEIVE[0] = (1<<PPIB_PUBLISH_RECEIVE_CHIDX_Pos) |
PPIB_PUBLISH_RECEIVE_EN_Msk;
   NRF_DPPIC30->CHENSET = DPPIC_CHENSET_CH1_Msk;

   // PERI PD
   NRF_PPIB21->PUBLISH_RECEIVE[0] = (5<<PPIB_PUBLISH_RECEIVE_CHIDX_Pos) |
PPIB_PUBLISH_RECEIVE_EN_Msk;
   NRF_PPIB22->SUBSCRIBE_SEND[0] = (5<<PPIB_SUBSCRIBE_SEND_CHIDX_Pos) |
PPIB_SUBSCRIBE_SEND_EN_Msk;
   NRF_DPPIC20->CHENSET = DPPIC_CHENSET_CH5_Msk;
```

6.3.4 Registers

Instances

Instance	Domain	Base address	Description
PPIB00	GLOBAL	0x40044000	PPI bridge PPIB00
PPIB01	GLOBAL	0x40045000	PPI bridge PPIB01
PPIB10	GLOBAL	0x40083000	PPI bridge PPIB10
PPIB11	GLOBAL	0x40084000	PPI bridge PPIB11
PPIB20	GLOBAL	0x400C3000	PPI bridge PPIB20
PPIB21	GLOBAL	0x400C4000	PPI bridge PPIB21
PPIB22	GLOBAL	0x400C5000	PPI bridge PPIB22
PPIB30	GLOBAL	0x40103000	PPI bridge PPIB30

Configuration

Instance	Domain	Configuration
PPIB00	GLOBAL	Bridges PPI channels 0-11 between PPIB_00 (MCU) and PPIB_10 (RADIO)
PPIB01	GLOBAL	Bridges PPI channels 0-7 between PPIB_01 (MCU) and PPIB_20 (PERI)
PPIB10	GLOBAL	Bridges PPI channels 0-11 between PPIB_10 (RADIO) and PPIB_00 (MCU)
PPIB11	GLOBAL	Bridges PPI channels 0-15 between PPIB_11 (RADIO) and PPIB_21 (PERI)
PPIB20	GLOBAL	Bridges PPI channels 0-7 between PPIB_20 (PERI) and PPIB_01 (MCU)
PPIB21	GLOBAL	Bridges PPI channels 0-15 between PPIB_21 (PERI) and PPIB_11 (RADIO)
PPIB22	GLOBAL	Bridges PPI channels 0-3 between PPIB_22 (PERI) and PPIB_30 (LP)
PPIB30	GLOBAL	Bridges PPI channels 0-3 between PPIB_30 (LP) and PPIB_22 (PERI)

Register overview

Register	Offset	Description
TASKS_SEND[n]	0x000	This task is unused, but the PPIB provides the SUBSCRIBE task to connect SEND [n] task.
SUBSCRIBE_SEND[n]	0x080	Subscribe configuration for task SEND[n]
EVENTS_RECEIVE[n]	0x100	This event is unused, but the PPIB provides the PUBLISH event to connect RECEIVE [n] event.
PUBLISH_RECEIVE[n]	0x180	Publish configuration for event RECEIVE[n]
OVERFLOW.SEND	0x400	The task overflow for SEND tasks using SUBSCRIBE_SEND. Write 0 to clear.

6.3.4.1 TASKS_SEND[n] (n=0..31)

Address offset: 0x000 + (n × 0x4)

This task is unused, but the PPIB provides the SUBSCRIBE task to connect SEND [n] task.

Writes to SEND [n] task are 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																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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			This task is unused, but the PPIB provides the SUBSCRIBE task to connect SEND [n] task. Writes to SEND [n] task are ignored.
			Trigger	1	Trigger task

6.3.4.2 SUBSCRIBE_SEND[n] (n=0..31)

Address offset: 0x080 + (n × 0x4)

Subscribe configuration for task SEND[n]

Writes to SEND [n] task are 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																									A	A	A	A	A	A	A	A
													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
A	RW	CHIDX		[0..255]	DPPI channel that task SEND[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.3.4.3 EVENTS_RECEIVE[n] (n=0..31)

Address offset: 0x100 + (n × 0x4)

This event is unused, but the PPIB provides the PUBLISH event to connect RECEIVE [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	RW	EVENTS_RECEIVE			This event is unused, but the PPIB provides the PUBLISH event to connect RECEIVE [n] event.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.3.4.4 PUBLISH_RECEIVE[n] (n=0..31)

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		[0..255]	DPPI channel that event RECEIVE[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.3.4.5 OVERFLOW.SEND

Address offset: 0x400

The task overflow for SEND tasks using SUBSCRIBE_SEND.

Write 0 to clear.

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

ID	R/W	Field	Value ID	Value	Description
A-f	RW	SEND[i] (i=0..31)			The status for tasks overflow at SUBSCRIBE_SEND[i].
			Overflow	1	Task overflow is happened.
			NoOverflow	0	Task overflow is not happened.

7 Security

The device is designed with basic security functions.

The main security features of the device are the following:

- Debug access protection
- Random number generator
- Immutable boot region for establishing root of trust

7.1 Security components

7.1.1 CRACEN TRNG — True Random Number Generator

The CRACEN TRNG provides true random number generation using the CRACEN peripheral, with only the random number generator functionality enabled.

To use the CRACEN TRNG, the CRACEN peripheral must first be configured.

After a reset, the TRNG function is disabled by default. Enable it by writing to the ENABLE register.

Once the TRNG is enabled through the CRACEN peripheral, configure and activate it using the RNGCONTROL.CONTROL register to begin random number generation.

7.1.1.1 Random Number Generator

The NIST-800-90B non-deterministic, true random number generator, can be used by cryptographic modules as an entropy source.

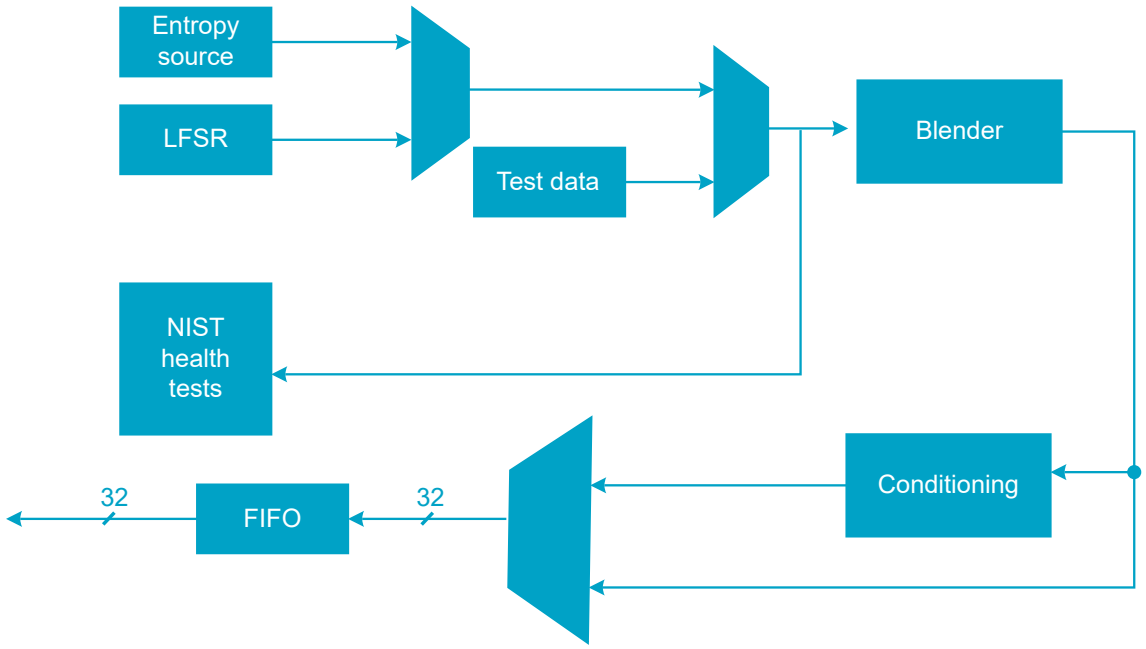


Figure 27: Random number generator block diagram

7.1.1.2 Entropy Source

The entropy source is a set of distinct oscillator rings, where each ring is composed of an odd number of inverters and oscillates at a much higher frequency than the digital sampling frequency. The analog signal from all rings is converted into a digital signal by sampling it at a regular clock frequency.

It is possible to observe the samples from this noise source by bypassing the conditioning function (by setting the bit CONTROL.CONDBYPASS to 1).

7.1.1.3 Blender

There are 4 shares (sets) of 8 distinct ring oscillators. The output bitstream from each set has its own set of tests as described in Continuous Testing on page 109. These bitstreams are merged by a blender to produce a single stream of entropy.

The most efficient way to generate random data with minimal latency is to concatenate the ring oscillator output streams (the hardware default). However, these outputs can be blended to increase entropy. The blender can be configured to use different methods for entropy enhancement via the BLENDINGMETHOD field. XORLEVEL1 reduces 4×8 bits to 4 bits, while XORLEVEL2 further reduces the 4 bits to 1. Another blending option is the Von-Neumann debiasing algorithm (VONNEUMANN), which assumes the input bits are independent and identically distributed (IID), producing a likely uniform output. When the Von-Neumann debiasing method is selected, the XORLEVEL1 and XORLEVEL2 compression methods are applied first, then 00 and 11 bit pairs are discarded while 10 and 01 pairs are retained, with only the first bit of each retained pair being output.

7.1.1.4 Sampling Process

When the output node of a high frequency ring oscillator (RO) is sampled at the rate of a low-frequency clock, then the sampled bit is random. The device ensures that $T_{osc} >> T_{RO}$, where T_{osc} is the duration during which the ring oscillator is free-running.

Two oscillation modes are supported: warm mode and cold mode.

In warm mode, the ring oscillators are free-running as long as the FIFO is being filled. Warm mode is selected by writing register COOLDOWNPERIOD = 0. A sample is taken every D_{OSC} clock cycles, where D_{OSC} is configured using the SAMPLINGPERIOD register. Note that the oscillation time $T_{OSC} = D_{OSC} * T_{CLK}$.

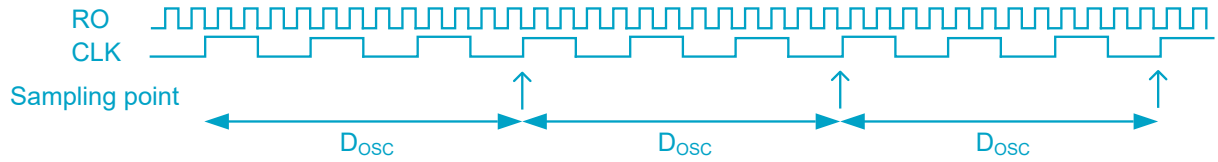


Figure 28: Sampling process - Warm mode

In cold mode, the ring oscillators are disabled for D_{RST} clock cycles after every sample. D_{RST} is configured through the COOLDOWNPERIOD register.

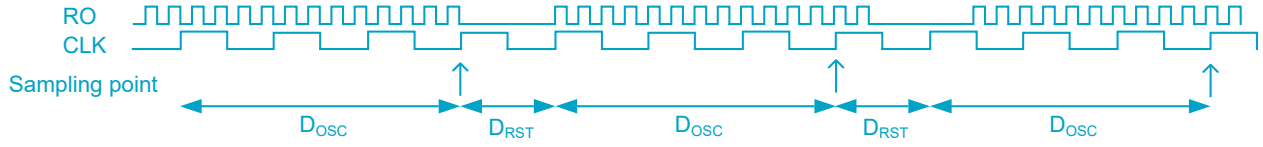


Figure 29: Sampling process - Cold mode

Note that in reality D_{OSC} is typically larger than what is suggested by the above figures, ensuring that $T_{OSC} >> T_{RST}$.

Warm mode could theoretically exhibit temporal correlations, unlike cold mode, which is rendered memoryless by resetting to a predefined state before each random data generation. To mitigate this in warm mode, an additional margin can be applied to T_{OSC}, by using the SAMPLINGPERIOD register. Health tests will activate if issues arise.

Cold mode may be more susceptible to side-channel attacks, as the cessation of oscillation offers a synchronization point, potentially allowing an attacker to more readily determine when samples are generated. Warm mode is not entirely immune to such vulnerabilities either, because the AES can produce a power signature that also serves as a synchronization point.

The warm mode will have a better throughput.

The choice between the two options depends on the application requirements. Both approaches are valid and pass the relevant test suites .

7.1.1.5 Continuous Testing

The samples from entropy source are monitored permanently by repetition and proportion tests running in parallel. Tests are run on the raw samples, i.e. not influenced by the CONTROL.BLENDINGMETHOD setting. Optionally, correlation and auto-correlation tests can be used as well.

7.1.1.5.1 Repetition Count Test

The repetition count test as described in section 4.4.1 of NIST Special Publication 800-90 is implemented. The acceptable false-positive rate defaults to $\alpha = 2^{-20}$.

For a min-entropy of H=8.0, the default repetition cut-off value (REPEATTHRESHOLD) is 4. Test can be disabled via DISREPETTESTS bit in the CONTROL register. Failures in the repetition test for each share are reported in the REPTESTFAILPERSHARE bits in the STATUS register.

7.1.1.5.2 Adaptive Proportion Test

The proportion count test as described in section 4.4.2 of NIST Special Publication 800-90 has been implemented with a window size of W = 512 non-binary samples. The acceptable false-positive rate has been fixed to $\alpha = 2^{-20}$. The entropy source delivers 1-bit sample and after chosen blending method non-binary samples are created. For a min-entropy of H=8.0, default proportion cut-off value is C = 1+CRITBINOM(W, 2^{-H},1-α) = 13. Cut-off value is programmed in PROPTESTCUTOFF register. This test can be disabled via CONTROL.DISPROPTESTS. Failing per share proportion test is reported via STATUS.PROPTESTFAILPERSHARE.

7.1.1.5.3 NIST-800-90B Start-up Test

Compliant with NIST-800-90B, the Repetition Count Test and the Adaptive Proportion Test are required to pass before any data is read from the FIFO.

7.1.1.5.4 Correlation Count Test

When two ring oscillators are coupled through injection locking, their bitstreams become correlated. Correlation count test can detect this occurrence. It uses time-division multiplexing and even single pair of ring oscillators case is supported. Window size of W = 128 samples is used with cut-off values defined in CORRTESTCUTOFF registers. Specific delay checks (or all checks) can be disabled via CONTROL.DISCORRTESTS.

7.1.1.5.5 Autocorrelation Count Test

When a ring oscillator is continuously sampled without accumulating enough jitter, the produced bitstream shows periodic behavior. Autocorrelation count test is able to detect this behavior. It uses time-division multiplexing and even single ring oscillator case is supported. Window size of W = 128 samples is used with cut-off values defined as in AUTOCORRTESTCUTOFF registers. Specific delay checks (or all checks) can be disabled via CONTROL.DISAUTOCORRTESTS.

7.1.1.6 Conditioning

The CBC-MAC mode (as described in section 3.1.5.1.1 and Appendix F in NIST Special Publication 800-90) of AES128 has been used as FIPS-approved keyed conditioning function. The conditioning function takes CONTROL.NB128BITBLOCKS x 128 bits from the entropy source as input and generates 128 bits as output. Assuming a min-entropy of 50% and NB128BITBLOCKS = 4, the input entropy is hin=512 x 50%=256 bits. According to section 3.1.5.1.2 in NIST Special Publication 800-90, the output of the conditioning function can be considered to have 128 bits of full entropy. The conditioning function can be tested by sending known data (using TESTDATA with CONTROL.TESTEN=1).

7.1.1.7 FIFO

The output entropy is split into 32-bit words and stored in the output FIFO. The FIFO size in 32-bit words can be read from the FIFODEPTH register. The FIFO operates as a single word stream - the first available word is returned regardless of which FIFO entry is accessed. When a word is read from the FIFO, the FIFOLEVEL register automatically decreases to reflect the number of remaining 32-bit random numbers. The FIFOTHRESHOLD register determines when the RNG state machine starts refilling the FIFO.

Depending on the configuration of the conditioning function, either raw noise samples or conditioned samples are written to the FIFO when the ENABLE bit is set. For details on the state machine that manages the filling of the FIFO, see Control FSM state diagram on page 111. If the CONTROL.FIFOWRITESTARTUP bit is set, the statemachine will start filling the FIFO immediately after the FIFO is enabled, ignoring startup failures. If the CONTROL.IGNOREHEALTHTESTSFAILFORFSM bit is set the state machine ignores all the health test failures and allows data to be written to the FIFO even if health tests fail. It is not recommended to set either of these bits, as it may lead to the FIFO being filled with data of poor quality.

Note:

When deviating from recommended hardware settings, it is the user's responsibility to verify the validity of the data in the FIFO.

7.1.1.8 Control FSM

The RNG is controlled by an finite state machine (FSM) which handles all the low-level interactions between the different hardware components described in Random number generator block diagram on page 107. During normal operation the FSM ensures that the entropy source is controlled in a NIST-800-90B compatible way, as shown in the following figure.

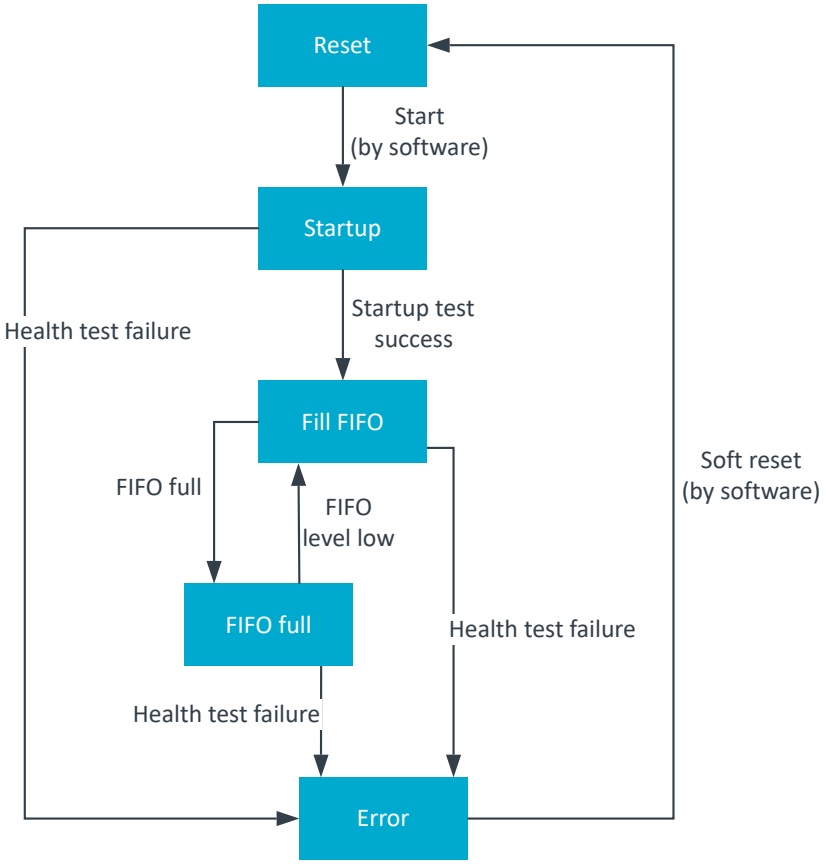


Figure 30: Control FSM state diagram

The state diagram does not show that asserting the hard or soft resets and/or de-asserting the enable bit cause the state to be reset to "Reset".

7.1.1.8.1 Check the Conditioning Function

1. Apply software reset (RNGCONTROL.CONTROL.SOFTRST = 1)
2. Write the control register
   - CONTROL.NB128BITBLOCKS=4
   - CONTROL.TESTEN=1
   - CONTROL.FIFOWRITESTARTUP=1
   - CONTROL.ENABLE=1
3. Write the KEY register
4. Write the data to the TESTDATA register (after each write, wait until the TESTDATABUSY flag becomes low)
5. Read the result from the FIFO

The table below shows an example of data (taken from section F.2.1 in NIST Special Publication 800-38A)

	128-bit format	32-bit APB format
Key	0x2B7E151628AED2A6ABF7158809CF4F3C	0x16157E2B 0xA6D2AE28 0x8815F7AB 0x3C4FCF09
Input	0x6BC0BCE12A459991E134741A7F9E1925	0xE1BCC06B 0x9199452A 0x1A7434E1 0x25199E7F
	0xAE2D8A571E03AC9C9EB76FAC45AF8E51	0x578A2DAE 0x9CAC031E 0xAC6FB79E 0x518EAF45
	0x30C81C46A35CE411E5FBC1191A0A52EF	0x461CC830 0x11E45CA3 0x19C1FBE5 0xEF520A1A
	0xF69F2445DF4F9B17AD2B417BE66C3710	0x45249FF6 0x179B4FDF 0x7B412BAD 0x10376CE6
Expected output	0x3FF1CAA1681FAC09120ECA307586E1A7	0xA1CAF13F 0x09AC1F68 0x30CA0E12 0xA7E18675

Table 22: Known-answer test for conditioning function

7.1.1.8.2 Check the entropy source

1. Apply software reset
2. Write the CONTROL register ensuring that none of the health tests are bypassed and set the ENABLE bit.
3. Wait until the STATE is not `Reset`, `Startup` or `Error`. When this is the case, all start-up tests have passed and the entropy source is working correctly.
4. If the state is `Error`, select different values for SAMPLINGPERIOD and WARMUPPERIOD and go back to 1.

7.1.1.8.3 Program the Key

1. Check the FIFOLEVEL register to monitor the amount of generated random numbers or wait for the FULLINT IRQ.
2. When FIFOLEVEL has reached the expected value or when an IRQ has been received, read the random numbers from the FIFO.
3. Use 4 x 32-bit random values to program a random key in the KEY register.
4. Apply software reset to flush the FIFO. By toggling the SOFTRST bit in the CONTROL register.

7.1.1.8.4 Get Samples

1. Check the FIFOLEVEL register to monitor the amount of generated random numbers or wait for the FIFOFULL IRQ.
2. When FIFOLEVEL has reached the expected value or when an IRQ has been received, read the random numbers from the FIFO.

7.1.1.8.5 Change Configuration

1. Disable the RNG (set the ENABLE bit of the CONTROL register to `0`)
2. Configure the CONTROL register
3. Do a soft reset to have a clean starting point (set bit SOFTRST of the CONTROL register to `1`)
4. Enable the RNG back (set the ENABLE bit of the CONTROL register to `1` while clearing the SOFTRST bit)

7.1.1.8.6 Data Format - Byte Ordering

All cryptographic data are handled following the big-endian format (AES standard). That means that the first byte (lowest address) of the data is the Most Significant Byte (MSB).

The APB bus uses little endian format. That means that the Least Significant Byte (LSB) is stored at the lowest address.

For example, a 128-bit block D[A] = 0x00112233445566778899AABBCCDDEEFF stored at address A (multiple of 4) appears on APB bus as the following 32-bit words:

- D[A+0] = 0x33221100
- D[A+4] = 0x77665544
- D[A+8] = 0xBBAA9988
- D[A+12] = 0xFFDDEEC

7.1.1.9 Performance

When conditioning is enabled, the number of cycles required for the RNG to generate the first *N* words of data can be estimated with the following formula:

*Nbcycles ≈ C + 1024 * (*SAMPLINGPERIOD *+* COOLDOWNPERIOD*) + 128 *** NB128BITBLOCKS *** max(1/8, (*SAMPLINGPERIOD *+* COOLDOWNPERIOD *+ 1) * BLENDING_RATIO / 128) * max(0, N - 4) +* WARMUPPERIOD

where *C* is the number of cycles required to initialize the RNG state machine.

BLENDING_RATIO is the ratio of bit reduction due to the chosen BLENDINGMETHOD (1 for CONCATENATION, 8 for XORLEVEL1, 32 for XORLEVEL2, and 128 for VONNEUMANN).

7.1.1.10 Interrupt

To enable the RNG interrupt, enable the RNG field in the CRACEN.INTEN register and set the necessary INTEN bits in the CRACENCORE.RNGCONTROL.CONTROL register.

The RNG has two interrupt causes:

- FULLINT: set when the FIFO becomes full. It can be cleared when FIFO is no longer full by writing to the FIFOLEVEL register or activating soft-reset.
- ANYHEALTHTESTFAIL: set when an error is detected in any of the health tests. It is cleared when writing a zero to the corresponding bit in the status register or when the soft-reset is activated.

For each interrupt, there is a status bit in the STATUS register and an enable bit in the CONTROL register. The STATUS register state does not depend on the interrupt enable bits in the control register. The interrupt enable bits are only used to control the generation of the event CRACEN.EVENTS_RNG and the signal to the interrupt handler.

7.1.1.11 Registers

Instances

Instance	Domain	Base address	Description
CRACEN	GLOBAL	0x40059000	True Random Number Generator

Register overview

Register	Offset	Description
EVENTS_RNG	0x104	Event indicating that interrupt triggered at RNG
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
ENABLE	0x400	Enable RNG peripheral module.

7.1.1.11.1 EVENTS_RNG

Address offset: 0x104

Event indicating that interrupt triggered at RNG

The interrupt source must be cleared at RNG before clearing this 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_RNG			Event indicating that interrupt triggered at RNG
					The interrupt source must be cleared at RNG before clearing this event.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.1.1.11.2 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RNG			Enable or disable interrupt for event RNG
					The interrupt source must be cleared at RNG before clearing this event.
			Disabled	0	Disable
			Enabled	1	Enable

7.1.1.11.3 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RNG			Write '1' to enable interrupt for event RNG
	W1S				The interrupt source must be cleared at RNG before clearing this event.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.1.1.11.4 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	RNG			Write '1' to disable interrupt for event RNG
	W1C				The interrupt source must be cleared at RNG before clearing this event.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.1.1.11.5 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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	RNG			Read pending status of interrupt for event RNG
					The interrupt source must be cleared at RNG before clearing this event.
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

7.1.1.11.6 ENABLE

Address offset: 0x400

Enable RNG peripheral 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	RNG			Enable RNG
			Disabled	0	RNG disabled.
			Enabled	1	RNG enabled.

7.1.1.12 Registers

Instances

Instance	Domain	Base address	Description
CRACENCORE	GLOBAL	0x40059000	True Random Number Generator

Configuration

Instance	Domain	Configuration
CRACENCORE	GLOBAL	RNGCONTROL registers included

Register overview

Register	Offset	Description
RNGCONTROL.CONTROL	0x500	Control register
RNGCONTROL.FIFOLEVEL	0x504	FIFO level register.
RNGCONTROL.FIFOTHRESHOLD	0x508	FIFO threshold register.
RNGCONTROL.FIFODEPTH	0x50C	FIFO depth register.
RNGCONTROL.KEY[n]	0x510	Key register.
RNGCONTROL.TESTDATA	0x520	Test data register.
RNGCONTROL.REPEATTHRESHOLD	0x524	Repetition test cut-off register.
RNGCONTROL.PROPTESTCUTOFF	0x528	Proportion test cut-off register.
RNGCONTROL.LFSRSEED	0x52C	LFSR seed register.
RNGCONTROL.STATUS	0x530	Status register.
RNGCONTROL.WARMUPPERIOD	0x534	Number of clock cycles in warm-up sequence.
RNGCONTROL.DISABLEOSC	0x538	DisableOsc register.
RNGCONTROL.SAMPLINGPERIOD	0x544	Number of clock cycles between sampling moments.
RNGCONTROL.HWCONFIG	0x558	Hardware configuration register.
RNGCONTROL.COOLDOWNPERIOD	0x55C	Number of clock cycles in cool-down sequence.
RNGCONTROL.AUTOCORRTESTCUTOFF0	0x560	AutoCorrTestCutoff register 0
RNGCONTROL.AUTOCORRTESTCUTOFF1	0x564	AutoCorrTestCutoff register 1
RNGCONTROL.CORRTESTCUTOFF0	0x568	CorrTestCutoff register 0
RNGCONTROL.CORRTESTCUTOFF1	0x56C	CorrTestCutoff register 1
RNGCONTROL.AUTOCORRTESTFAILED	0x570	Auto-correlation test failing ring(s).
RNGCONTROL.CORRTESTFAILED	0x574	Correlation test failing ring.
RNGCONTROL.HWVERSION	0x57C	Fixed to 1 for this version.
RNGCONTROL.FIFO[n]	0x580	FIFO data

7.1.1.12.1 RNGCONTROL.CONTROL

Address offset: 0x500

Control register

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	P	P	O	O	O			N	N	M	L	K	J	J	J	J				I	H			G	F			E	D	C	B	A
Reset 0x00040000	**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**	**0**

ID	R/W	Field	Value ID	Value	Description
A	W	ENABLE			Start the NDRNG. Self-clearing bit.

Bit number				31	30	29	28	27	26	25	24	23	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	P	O	O	O			N	N	M	L	K	J	J	J	J				I	H			G	F			E	D	C	B	A
Reset 0x00040000				**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**	**0**
ID	R/W	Field	Value ID	Value																															
B	RW	LFSREN																																	
																																			Select between the NDRNG with asynchronous free running oscillators (when 0) and the Pseudo-Random generator with synchronous oscillators for simulation purpose (when 1).
C	RW	TESTEN																																	Select input for conditioning function and continuous tests:
			NORMAL	0																															Noise source (normal mode).
			TEST	1																															Test data register (test mode).
D	RW	CONDBYPASS																																	Conditioning function bypass.
			NORMAL	0																															the conditioning function is used (normal mode).
			BYPASS	1																															the conditioning function is bypassed (to observe entropy source directly).
E	RW	INTENREP																																	Enable interrupt if any of the health test fails.
F	RW	INTENFULL																																	Enable interrupt if FIFO is full.
G	RW	SOFTRST																																	Datapath content flush and control FSM
H	RW	FORCEACTIVEROS																																	Force oscillators to run when FIFO is full.
I	RW	IGNOREHEALTHTESTSFAILFORFSM																																	Results of the health tests during start-up and online test do not affect the control FSM state. It also bypass control FSM StartUp phase.
J	RW	NB128BITBLOCKS																																	Number of 128 bit blocks used in conditioning (AES-CBC-MAC) post-processing. Zero value is not allowed.
K	RW	FIFOWRITESTARTUP																																	Enable write of the samples in the FIFO during start-up.
L	RW	DISREPETTESTS																																	All repetition tests (each share) are disabled via this single bit.
M	RW	DISPROPTESTS																																	All proportion tests (each share) are disabled via this single bit.
N	RW	DISAUTOCORRTESTS																																	Disable specific delay(s) check in auto-correlation test - same RO: x1: vs. the following sample of the same RO -> (0) delay check 1x: vs. the later sample of the same RO -> (+1) delay check
O	RW	DISCORRTESTS																																	Disable specific delay(s) check in correlation test - different ROs: xx1: vs. the same sample of the other RO -> (0) delay check x1x: vs. the preceding sample of the other RO -> (-1) delay check 1xx: vs. the next sample of the other RO -> (+1) delay check
P	RW	BLENDINGMETHOD																																	Select blending method
			CONCATENATION	0																															Concatenation
			XORLEVEL1	1																															XOR level 1
			XORLEVEL2	2																															XOR level 2
			VONNEUMANN	3																															VON-NEUMANN debiasing

7.1.1.12.2 RNGCONTROL.FIFOLEVEL

Address offset: 0x504

FIFO level 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	FIFOLEVEL			Number of 32 bits words of random values available in the FIFO. Any write to this register clears the FULLINT flag in the STATUS register, but does not affect this register content. Note that if the FIFO is still full when writing this register, the status flag and interrupt will be set back up right away

7.1.1.12.3 RNGCONTROL.FIFOTHRESHOLD

Address offset: 0x508

FIFO threshold 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 0x00000003	**0**	**0**	**0**	**0**	**0**	**0**	**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**

ID	R/W	Field	Value ID	Value	Description
A	RW	FIFOTHRESHOLD			FIFO level threshold below which the module leaves the idle state to refill the FIFO. Expressed in number of 128bit blocks. After a FIFO read, the RNG will start refilling the FIFO if FIFOLEVEL is smaller than (FIFOTHRESHOLD + 1) * 4

7.1.1.12.4 RNGCONTROL.FIFODEPTH

Address offset: 0x50C

FIFO depth 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 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	R	FIFODEPTH			Maximum number of 32 bits words that can be stored in the FIFO.

7.1.1.12.5 RNGCONTROL.KEY[n] (n=0..3)

Address offset: 0x510 + (n × 0x4)

Key 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	KEY			Key register. Key0 is byte 0-3, Key1 is byte 4-7, Key2 is byte 8-11 and Key3 is byte 12-15.

7.1.1.12.6 RNGCONTROL.TESTDATA

Address offset: 0x520

Test data register.

This register is used to feed known data to the conditioning function or to the continuous tests. When one word is written into this register, the 32-bit are sent to those modules. Since some time is needed for processing, there is one busy flag (TESTDATABUSY in the STATUS register) going high as soon as data is written, and going low when the next word can be written. Write access to this register is ignored when CONTROL.TESTEN is 0. Test data written through this interface is expected to be a multiple of 128 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 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	W	TESTDATA																																	
				Test data register.																															

7.1.1.12.7 RNGCONTROL.REPEATTHRESHOLD

Address offset: 0x524

Repetition test cut-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	A	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																															
A	RW	REPEATTHRESHOLD																																	
				Repetition Test cut-off value.																															

7.1.1.12.8 RNGCONTROL.PROPTESTCUTOFF

Address offset: 0x528

Proportion test cut-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	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																															
A	RW	PROPTESTCUTOFF																																	
				Proportion test cut-off value.																															

7.1.1.12.9 RNGCONTROL.LFSRSEED

Address offset: 0x52C

LFSR seed 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00FFFFFF				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	R/W	Field	Value ID	Value																															
A	RW	LFSRSEED																																	
				LFSR initialization value.																															
B	W	LFSRSSELECTION																																	
				Share index for which initialization value should be used.																															

7.1.1.12.10 RNGCONTROL.STATUS

Address offset: 0x530

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				J I I I I H H H H G F E D C B B B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	R	TESTDATABUSY			High when data written to TestData register is being processed.
B	R	STATE			State of the control FSM:
			RESET	0	Reset
			STARTUP	1	Startup
			IDLERON	2	Idle / FIFO full
			FILLFIFO	4	Fill FIFO
			ERROR	5	Error
C	R	REPFAIL			NIST repetition test(s) failure.
D	R	PROPFAIL			NIST proportion test(s) failure.
E	RW W0C	ANYHEALTHTESTFAIL			Any of the enabled health tests is failing.
F	R	FULLINT			FIFO full status.
G	RW W0C	STARTUPFAIL			Start-up test(s) failure.
H	R	REPTESTFAILPERSHARE			NIST Repetition test failure per share.
I	R	PROPTESTFAILPERSHARE			NIST Proportion test failure per share.
J	RW W0C	CONDITIONINGISTOOSLOW			Conditioning consumes data slower than they are provided to it. This can happen for SAMPLINGPERIOD < 15, BLENDINGMETHOD = CONCATENATION, or four shares.

7.1.1.12.11 RNGCONTROL.WARMUPPERIOD

Address offset: 0x534

Number of clock cycles in warm-up 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
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	WARMUPPERIOD			Number of clock cycles in warm-up sequence.

7.1.1.12.12 RNGCONTROL.DISABLEOSC

Address offset: 0x538

DisableOsc 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	DISABLEOSC			Disable oscillator rings. Oscillators are grouped in shares. Each group of 8 bits controls one share: bits [7:0] disable oscillators from the first share, bits [15:8] disable oscillators from the second share, bits [23:16] disable oscillators from the third share, and bits [31:24] disable oscillators from the fourth share.

7.1.1.12.13 RNGCONTROL.SAMPLINGPERIOD

Address offset: 0x544

Number of clock cycles between sampling moments.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 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 0x00000FFF				**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	SAMPLINGPERIOD			Number of clock cycles between sampling moments.

7.1.1.12.14 RNGCONTROL.HWCONFIG

Address offset: 0x558

Hardware 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				D D C C C C B B B A A A A A A A
Reset 0x0002410F				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	R	NBOFINV			Generic g_NbOfInverters value.
B	R	LOG2NBOFAUTOCORRTESTSPERSHARE			Generic g_Log2NbOfAutoCorrTestsPerShare value.
C	R	LOG2FIFODEPTH			Generic g_Log2FifoDepth value.
D	R	LOG2NBOFSHARES			Generic g_Log2NbOfShares value.

7.1.1.12.15 RNGCONTROL.COOLDOWNPERIOD

Address offset: 0x55C

Number of clock cycles in cool-down 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	COOLDOWNPERIOD			Number of clock cycles in cool-down sequence.

7.1.1.12.16 RNGCONTROL.AUTOCORRTESTCUTOFF0

Address offset: 0x560

AutoCorrTestCutoff 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				B B B B B B B A A A A A A A A
Reset 0x007F007F				**0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	DLYZEROCUTOFF			Auto-correlation test cut-off value for delay of 0 samples.
B	RW	DLYONECUTOFF			Auto-correlation test cut-off value for delay of +1 sample.

7.1.1.12.17 RNGCONTROL.AUTOCORRTESTCUTOFF1

Address offset: 0x564

AutoCorrTestCutoff 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				B B B B B B B A A A A A A A
Reset 0x007F007F				**0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	DLYTWOCUTOFF			Auto-correlation test cut-off value for delay of +2 samples.
B	RW	DLYTHREECUTOFF			Auto-correlation test cut-off value for delay of +3 samples.

7.1.1.12.18 RNGCONTROL.CORRTESTCUTOFF0

Address offset: 0x568

CorrTestCutoff 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				B B B B B B B A A A A A A A
Reset 0x007F007F				**0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	DLYZEROCUTOFF			Correlation test cut-off value for delay of 0 samples.
B	RW	DLYONECUTOFF			Correlation test cut-off value for delay of +/-1 sample.

7.1.1.12.19 RNGCONTROL.CORRTESTCUTOFF1

Address offset: 0x56C

CorrTestCutoff 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				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
Reset 0x007F007F				**0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	DLYTWOCUTOFF			Correlation test cut-off value for delay of +/- 2 samples.
B	RW	DLYTHREECUTOFF			Correlation test cut-off value for delay of +/- 3 samples.

7.1.1.12.20 RNGCONTROL.AUTOCORRTESTFAILED

Address offset: 0x570

Auto-correlation test failing ring(s).

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A 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	AUTOCORRTESTFAILED			Auto-correlation test failing ring(s).

7.1.1.12.21 RNGCONTROL.CORRTESTFAILED

Address offset: 0x574

Correlation test failing ring.

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

7.1.1.12.22 RNGCONTROL.HWVERSION

Address offset: 0x57C

Fixed to 1 for this version.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	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	R	HWVERSION			Fixed to 1 for this version.

7.1.1.12.23 RNGCONTROL.FIFO[n] (n=0..31)

Address offset: 0x580 + (n × 0x4)

FIFO data

The FIFO contains the RNG output 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 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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			FIFO data

7.1.2 TAMPC — Tamper controller

The tamper controller peripheral handles input from internal and external physical attack detectors and controls the device response.

The following figure shows an overview of the TAMPC detectors, input, and output.

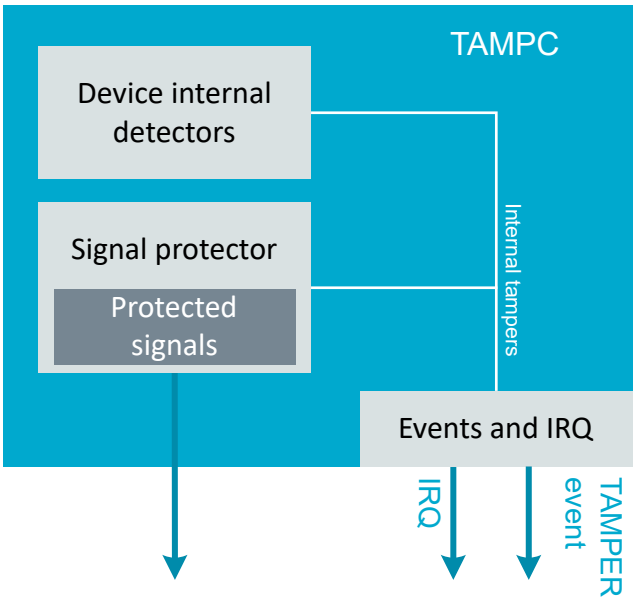


Figure 31: TAMPC overview

TAMPC implements the following physical security features:

- Detection of fault injection attacks (voltage glitching, electromagnetic fault injection, etc.)
  - Signal protector to guard critical configuration signals
  - Glitch detectors to detect timing violations of internal logic

The tamper detectors are divided into two categories: external and internal. The external detectors rely on external stimuli through dedicated GPIO pins, and the internal detectors rely on internal signals not exposed outside the device package.

Internal tamper detectors

A tamper attack detected by any of the internal tamper detectors indicates that the device's internal logic could be affected by the attack, and the system state could be compromised.

7.1.2.1 Glitch detector

The device implements general detectors to prevent fault injection attacks.

Detectors are strategically placed among the digital logic to detect local timing glitches (timing violations). The glitch detectors monitor the effects of attempted fault injection attacks and not the attack attempt itself. For example, fault injection attempts could be utilizing voltage glitches on supply or decoupling pins, or electro magnetic fault injection (EMFI) techniques. The detectors are designed and tuned to be more sensitive to timing violations than normal logic, enabling the digital logic to react before an injected fault is propagated through the system.

The glitch detectors are enabled from reset and can be disabled for debugging using the registers PROTECT.GLITCHSLOWDOMAIN.CTRL on page 133 and PROTECT.GLITCHFASTDOMAIN.CTRL on page 134.

When an internal tamper event is detected, one of the following will occur:

- A TAMPC EVENT is generated.
- A chip reset is triggered and the `SECTAMPER` value in the reset reason register indicates what happened, depending on the status of the register PROTECT.INTRESETEN.CTRL on page 135.

The internal tamper reset enable signal INTRESETEN is enabled from reset.

7.1.2.2 Signal protector

The device implements detectors to protect selected signals that control critical device features.

The signal protector implements one detector per protected signal to detect unintentional value changes in that signal. The detector notifies TAMPC if a protected signal changes value caused by tampering. The detectors are enabled from reset and can be disabled using the register , which is for debugging purposes only. A detected unintentional value change in any of the protected signals leads to an internal tamper event where one of the following occur:

- A TAMPC event is generated.
- A chip reset is triggered and the `SECTAMPER` value in the reset reason register indicates what happened, depending on the status of the register PROTECT.INTRESETEN.CTRL on page 135.

The INTRESETEN register is enabled from reset. The PROTECT.<component>.STATUS registers indicate which protected signal had an unintentional value change when the register INTENRESETEN is disabled.

The signal protector implements a two stage write cycle to change the value of a protected signal, in addition to a required write key which must be included for all register writes. The two stages are the following:

1. Initial register write to clear the write protection.
2. Register write to change the signal's value.

Write `Clear` to the WRITEPROTECTION field in the PROTECT.<component>.CTRL register to clear the write protection in the first register write. Then write to the VALUE and LOCK fields in the next register write operation.

> **Note:** It is required to clear the WRITEPROTECTION field before any updates to the VALUE and LOCK fields are accepted by the register.

The write protection is automatically re-enabled after the subsequent write to change the VALUE field when the register write does not include the `Clear` value in the WRITEPROTECTION field.

The LOCK field controls a lock feature which prevents further updates to the VALUE and LOCK fields until a reset with the required reset source for the specific signal is issued.

A WRITEERROR event is generated for any of the following conditions:

- Register write does not have the correct write key
- Write protection is active and the write operation does not contain the value to clear the write protection
- The lock is enabled

The sequence to change the VALUE or LOCK fields in the PROTECT.<component>.CTRL registers is as follows:

1. Write `Clear` to the WRITEPROTECTION field and `KEY` to the KEY field.
2. Write `Disabled` to the WRITEPROTECTION field, `KEY` to the KEY field, and `KEY` to the desired LOCK and VALUE fields.

7.1.2.2.1 Debugger signals

TAMPC provides protection for the following Arm CoreSight™ debugger signals.

Signal	Name	Notes
NIDEN	Non-Invasive Debug Enable	ETM/ITM trace and other non-halting debug methods
DBGEN	Invasive Debug Enable	The debugger may halt the CPU for debug purposes

Table 23: TAMPC protected debugger signals

See the appropriate*Arm CoreSight Technical Reference manual* for details on these signals.

For more information about using the protected debugger signals, see Debug on page 526.

7.1.2.3 TAMPC reset behavior

TAMPC registers are reset by different sources.

Protected signals in TAMPC are divided into the following reset source categories.

- Category 1 – Brownout reset, power-on reset
- Category 2 – Pin reset and TAMPC reset
- Category 3 – Watchdog timer reset, CPU lockup reset, System reset request, and Wake-up from System OFF reset

For more information about reset sources, see RESET — Reset control on page 88.

			Reset source		
Register	**Function**	**Reset value**	**Cat 1**	**Cat 2**	**Cat 3**
PROTECT.DOMAIN[0].DBGEN	Allow invasive debugging.	0	x	x	
PROTECT.DOMAIN[0].NIDEN	Allow non-invasive debugging.	0	x	x	
PROTECT.GLITCHSLOWDOMAIN	Enable slow domain glitch detector.	1	x	x	x
PROTECT.GLITCHFASTDOMAIN	Enable fast domain glitch detector.	1	x	x	x
PROTECT.EXTRESETEN	Enable automatic reset from external tamper detectors events.	0	x	x	x
PROTECT.INTRESETEN	Enable automatic reset from internal tamper detector events.	1	x	x	x
PROTECT.ERASEPROTECT	Allow device erase using CTRL-AP and RRAMC.	0	x	x	x

Table 24: TAMPC protected signals

7.1.2.4 Registers

Instances

Instance	Domain	Base address	Description
TAMPC	GLOBAL	0x400EF000	Tamper controller TAMPC

Configuration

Instance	Domain	Configuration
TAMPC	GLOBAL	Device can use SWD pins as GPIO pins. Reset value of field VALUE in register PROTECT.INTRESETEN.CTRL: 1

Register overview

Register	Offset	Description
EVENTS_TAMPER	0x100	Tamper controller detected an error.
EVENTS_WRITEERROR	0x104	Attempt to write a VALUE in PROTECT registers without clearing the WRITEPROTECT.
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
STATUS	0x400	The tamper controller status.
PROTECT.DOMAIN[n].DBGEN.CTRL	0x500	Control register for invasive (halting) debug enable for the local debug components within domain n.
PROTECT.DOMAIN[n].DBGEN.STATUS	0x504	Status register for invasive (halting) debug enable for domain n.
PROTECT.DOMAIN[n].NIDEN.CTRL	0x508	Control register for non-invasive debug enable for the local debug components within domain n.
PROTECT.DOMAIN[n].NIDEN.STATUS	0x50C	Status register for non-invasive debug enable for domain n.
PROTECT.AP[n].DBGEN.CTRL	0x700	Control register to enable invasive (halting) debug in domain ns access port.
PROTECT.AP[n].DBGEN.STATUS	0x704	Status register for invasive (halting) debug enable for domain ns access port.
PROTECT.GPIOSWD.CTRL	0x878	Control register for activating GPIO functionality.
PROTECT.GPIOSWD.STATUS	0x87C	Status register for active shield detector enable signal.
PROTECT.GLITCHSLOWDOMAIN.CTRL	0x940	Control register for slow domain glitch detectors enable signal.
PROTECT.GLITCHSLOWDOMAIN.STATUS	0x944	Status register for slow domain glitch detectors enable signal.
PROTECT.GLITCHFASTDOMAIN.CTRL	0x948	Control register for fast domain glitch detectors enable signal.
PROTECT.GLITCHFASTDOMAIN.STATUS	0x94C	Status register for fast domain glitch detectors enable signal.
PROTECT.INTRESETEN.CTRL	0x978	Control register for internal tamper reset enable signal.
PROTECT.INTRESETEN.STATUS	0x97C	Status register for internal tamper reset enable signal.
PROTECT.ERASEPROTECT.CTRL	0x980	Control register for erase protection.
PROTECT.ERASEPROTECT.STATUS	0x984	Status register for eraseprotect.

7.1.2.4.1 EVENTS_TAMPER

Address offset: 0x100

Tamper controller detected 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_TAMPER			Tamper controller detected an error.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.1.2.4.2 EVENTS_WRITEERROR

Address offset: 0x104

Attempt to write a VALUE in PROTECT registers without clearing the WRITEPROTECT.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_WRITEERROR			Attempt to write a VALUE in PROTECT registers without clearing the WRITEPROTECT.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

7.1.2.4.3 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	TAMPER			Enable or disable interrupt for event TAMPER
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	WRITEERROR			Enable or disable interrupt for event WRITEERROR
			Disabled	0	Disable
			Enabled	1	Enable

7.1.2.4.4 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 W1S	TAMPER			Write '1' to enable interrupt for event TAMPER
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	WRITEERROR			Write '1' to enable interrupt for event WRITEERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.1.2.4.5 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 W1C	TAMPER			Write '1' to disable interrupt for event TAMPER
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	WRITEERROR			Write '1' to disable interrupt for event WRITEERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

7.1.2.4.6 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	TAMPER			Read pending status of interrupt for event TAMPER
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B	R	WRITEERROR			Read pending status of interrupt for event WRITEERROR
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

7.1.2.4.7 STATUS

Address offset: 0x400

The tamper controller status.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to it.

Note: The glitch detectors must be reset using their CTRL registers before the STATUS register bits for glitch detectors can be cleared. The glitch detector continuously drives its output status signal to the STATUS register, hence clearing only the STATUS register is not sufficient.

Bit number				31 30 29 28 27 26 25 24 23 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 W1C	PROTECT			Error detected for the protected signals.
			NotDetected	0	Not detected.
			Detected	1	Detected.
B	RW W1C	GLITCHSLOWDOMAIN[i] (i=0..0)			Slow domain glitch detector i detected an error.
			NotDetected	0	Not detected.
			Detected	1	Detected.
C-F	RW W1C	GLITCHFASTDOMAIN[i] (i=0..3)			Fast domain glitch detector i detected an error.
			NotDetected	0	Not detected.
			Detected	1	Detected.

7.1.2.4.8 PROTECT.DOMAIN[n].DBGEN.CTRL (n=0..0)

Address offset: 0x500 + (n × 0x20)

Control register for invasive (halting) debug enable for the local debug components within 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				D D D D D D D D D D D D D D D D C C C C B 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 0 1 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Set value of dbgen signal.
			Low	0	Signal is logic 0, indicating that invasive debug is disabled.
			High	1	Signal is logic 1, indicating that invasive debug is enabled.
B	W1 W1S	LOCK			Lock this register to prevent changes to the VALUE field until next reset.
			Disabled	0	Lock disabled.
			Enabled	1	Lock enabled.
C	RW	WRITEPROTECTION			The write protection must be cleared to allow updates to the VALUE field. The write protection is cleared by writing CLEAR in a separate write operation prior to updating the VALUE and LOCK fields. The write protection is automatically enabled after the corresponding change to the VALUE field.
			Disabled	0x0	Read: Write protection is disabled.
			Enabled	0x1	Read: Write protection is enabled.
			Clear	0xF	Write: Value to clear write protection.
D	W	KEY			Required write key for upper 16 bits. Must be included in all register write operations.
			KEY	0x50FA	Write key value.

7.1.2.4.9 PROTECT.DOMAIN[n].DBGEN.STATUS (n=0..0)

Address offset: 0x504 + (n × 0x20)

Status register for invasive (halting) debug enable for domain n.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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	RW	ERROR			Error detection status.
	W1C				
			NoError	0	No error detected.
			Error	1	Error detected.

7.1.2.4.10 PROTECT.DOMAIN[n].NIDEN.CTRL (n=0..0)

Address offset: 0x508 + (n × 0x20)

Control register for non-invasive debug enable for the local debug components within 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				D D D D D D D D D D D D D D D D C C C C B 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 0 1 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Set value of niden signal.
			Low	0	Signal is logic 0, indicating that non-invasive debug is disabled.
			High	1	Signal is logic 1, indicating that non-invasive debug is enabled.
B	W1	LOCK			Lock this register to prevent changes to the VALUE field until next reset.
	W1S				
			Disabled	0	Lock disabled.
			Enabled	1	Lock enabled.
C	RW	WRITEPROTECTION			The write protection must be cleared to allow updates to the VALUE field. The write protection is cleared by writing CLEAR in a separate write operation prior to updating the VALUE and LOCK fields. The write protection is automatically enabled after the corresponding change to the VALUE field.
			Disabled	0x0	Read: Write protection is disabled.
			Enabled	0x1	Read: Write protection is enabled.
			Clear	0xF	Write: Value to clear write protection.
D	W	KEY			Required write key for upper 16 bits. Must be included in all register write operations.
			KEY	0x50FA	Write key value.

7.1.2.4.11 PROTECT.DOMAIN[n].NIDEN.STATUS (n=0..0)

Address offset: 0x50C + (n × 0x20)

Status register for non-invasive debug enable for domain n.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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	RW	ERROR			Error detection status.
	W1C				
			NoError	0	No error detected.
			Error	1	Error detected.

7.1.2.4.12 PROTECT.AP[n].DBGEN.CTRL (n=0..0)

Address offset: 0x700 + (n × 0x10)

Control register to enable invasive (halting) debug in domain ns access 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				D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D									C	C	C	C			B	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																															
A	RW	VALUE																																	
			Low	0																															
			High	1																															
B	W1	LOCK																																	
	W1S																																		
			Disabled	0																															
			Enabled	1																															
C	RW	WRITEPROTECTION																																	
			Disabled	0x0																															
			Enabled	0x1																															
			Clear	0xF																															
D	W	KEY																																	
			KEY	0x50FA																															

Write key value.

7.1.2.4.13 PROTECT.AP[n].DBGEN.STATUS (n=0..0)

Address offset: 0x704 + (n × 0x10)

Status register for invasive (halting) debug enable for domain ns access port.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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																															
A	RW	ERROR																																	
	W1C																																		
			NoError	0																															
			Error	1																															

Error detected.

7.1.2.4.14 PROTECT.GPIOSWD

Enable GPIO functionality on SWD pins.

The SWD pins can be used as GPIO pins.

7.1.2.4.14.1 PROTECT.GPIOSWD.CTRL

Address offset: 0x878

Control register for activating GPIO functionality.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D									C	C	C	C			B	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																															
A	RW	VALUE																																	
			Low	0																															
			High	1																															
B	W1	LOCK																																	
	W1S																																		
			Disabled	0																															
			Enabled	1																															
C	RW	WRITEPROTECTION																																	
			Disabled	0x0																															
			Enabled	0x1																															
			Clear	0xF																															
D	W	KEY																																	
			KEY	0x50FA																															

7.1.2.4.14.2 PROTECT.GPIOSWD.STATUS

Address offset: 0x87C

Status register for active shield detector enable signal.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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																															
A	RW	ERROR																																	
	W1C																																		
			NoError	0																															
			Error	1																															

7.1.2.4.15 PROTECT.GLITCHSLOWDOMAIN

Enable slow domain glitch detectors.

7.1.2.4.15.1 PROTECT.GLITCHSLOWDOMAIN.CTRL

Address offset: 0x940

Control register for slow domain glitch detectors enable 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				D D D D D D D D D D D D D D D D C C C C B A
Reset 0x00000011				**0 0 0 0 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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Set value of slow domain glitch detectors enable signal.
			Low	0	Signal is logic 0.
			High	1	Signal is logic 1.
B	W1 W1S	LOCK			Lock this register to prevent changes to the VALUE field until next reset.
			Disabled	0	Lock disabled.
			Enabled	1	Lock enabled.
C	RW	WRITEPROTECTION			The write protection must be cleared to allow updates to the VALUE field. The write protection is cleared by writing CLEAR in a separate write operation prior to updating the VALUE and LOCK fields. The write protection is automatically enabled after the corresponding change to the VALUE field.
			Disabled	0x0	Read: Write protection is disabled.
			Enabled	0x1	Read: Write protection is enabled.
			Clear	0xF	Write: Value to clear write protection.
D	W	KEY			Required write key for upper 16 bits. Must be included in all register write operations.
			KEY	0x50FA	Write key value.

7.1.2.4.15.2 PROTECT.GLITCHSLOWDOMAIN.STATUS

Address offset: 0x944

Status register for slow domain glitch detectors enable signal.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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	RW W1C	ERROR			Error detection status.
			NoError	0	No error detected.
			Error	1	Error detected.

7.1.2.4.16 PROTECT.GLITCHFASTDOMAIN

Enable fast domain glitch detectors.

7.1.2.4.16.1 PROTECT.GLITCHFASTDOMAIN.CTRL

Address offset: 0x948

Control register for fast domain glitch detectors enable 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				D D D D D D D D D D D D D D D D C C C C B A
Reset 0x00000011				**0 0 0 0 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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Set value of fast domain glitch detector's enable 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				D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D									C	C	C	C			B	A
Reset 0x00000011				**0**	**0**	**0**	**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**	**1**
ID	R/W	Field	Value ID	Value																															
			Low	0																															
			High	1																															
B	W1	LOCK																																	
	W1S																																		
			Disabled	0																															
			Enabled	1																															
C	RW	WRITEPROTECTION																																	
			Disabled	0x0																															
			Enabled	0x1																															
			Clear	0xF																															
D	W	KEY																																	
			KEY	0x50FA																															

7.1.2.4.16.2 PROTECT.GLITCHFASTDOMAIN.STATUS

Address offset: 0x94C

Status register for fast domain glitch detectors enable signal.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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																															
A	RW	ERROR																																	
	W1C																																		
			NoError	0																															
			Error	1																															

7.1.2.4.17 PROTECT.INTRESETEN

Trigger a reset when tamper is detected by the glitch detectors, signal protector or CRACEN tamper detector.

This reset gives reset reason *SECTAMPER*

7.1.2.4.17.1 PROTECT.INTRESETEN.CTRL

Address offset: 0x978

Control register for internal tamper reset enable 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				D D D D D D D D D D D D D D D D C C C C B A
Reset 0x00000011				**0 0 0 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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	VALUE			Set value of internal tamper reset enable signal.
			Low	0	Signal is logic 0.
			High	1	Signal is logic 1.
B	W1 W1S	LOCK			Lock this register to prevent changes to the VALUE field until next reset.
			Disabled	0	Lock disabled.
			Enabled	1	Lock enabled.
C	RW	WRITEPROTECTION			The write protection must be cleared to allow updates to the VALUE field. The write protection is cleared by writing CLEAR in a separate write operation prior to updating the VALUE and LOCK fields. The write protection is automatically enabled after the corresponding change to the VALUE field.
			Disabled	0x0	Read: Write protection is disabled.
			Enabled	0x1	Read: Write protection is enabled.
			Clear	0xF	Write: Value to clear write protection.
D	W	KEY			Required write key for upper 16 bits. Must be included in all register write operations.
			KEY	0x50FA	Write key value.

7.1.2.4.17.2 PROTECT.INTRESETEN.STATUS

Address offset: 0x97C

Status register for internal tamper reset enable signal.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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	RW W1C	ERROR			Error detection status.
			NoError	0	No error detected.
			Error	1	Error detected.

7.1.2.4.18 PROTECT.ERASEPROTECT

Device erase protection.

7.1.2.4.18.1 PROTECT.ERASEPROTECT.CTRL

Address offset: 0x980

Control register for 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				D D D D D D D D D D D D D D D D C C C C B 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	VALUE			Set value of eraseprotect 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				D D D D D D D D D D D D D D D D C C C C B 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
			Low	0	Signal is logic 0.
			High	1	Signal is logic 1.
B	W1 W1S	LOCK			Lock this register to prevent changes to the VALUE field until next reset.
			Disabled	0	Lock disabled.
			Enabled	1	Lock enabled.
C	RW	WRITEPROTECTION			The write protection must be cleared to allow updates to the VALUE field. The write protection is cleared by writing CLEAR in a separate write operation prior to updating the VALUE and LOCK fields. The write protection is automatically enabled after the corresponding change to the VALUE field.
			Disabled	0x0	Read: Write protection is disabled.
			Enabled	0x1	Read: Write protection is enabled.
			Clear	0xF	Write: Value to clear write protection.
D	W	KEY			Required write key for upper 16 bits. Must be included in all register write operations.
			KEY	0x50FA	Write key value.

7.1.2.4.18.2 PROTECT.ERASEPROTECT.STATUS

Address offset: 0x984

Status register for eraseprotect.

Note: Unless cleared, the STATUS register will be cumulative. A field is cleared by writing '1' to 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	RW W1C	ERROR			Error detection status.
			NoError	0	No error detected.
			Error	1	Error detected.

8 Peripherals

The device features a rich set of peripherals. The following sections describe the peripherals and how they are used.

8.1 Peripheral interface

Peripherals are controlled by the CPU through configuration, 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 or the CPU by tying events to CPU interrupts.

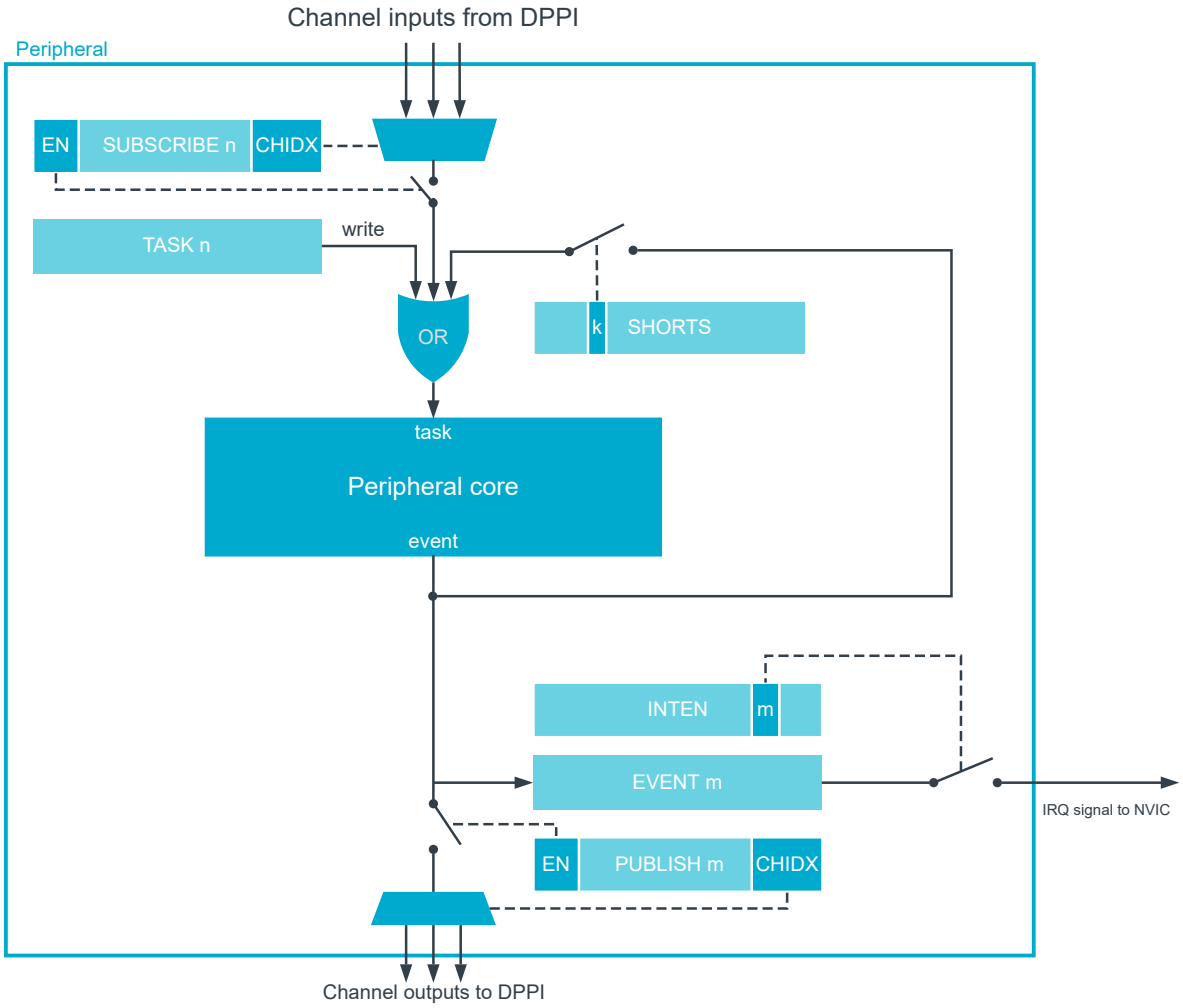


Figure 32: Peripheral interface

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 94.

8.1.2 Peripheral ID

Each peripheral is assigned a fixed block of address space that is minimum 4 KB in size and has at least 1024 registers of 32 bits.

For more information on available peripherals and their location in the address map, see Instantiation on page 141.

There is a direct relationship between peripheral ID and base address:

```
base_address = 0x40000000 + 0x1000 * ID
```

Example peripheral base addresses:

- 0x40000000 is assigned ID=0
- 0x40001000 is assigned ID=1
- 0x4001F000 is assigned ID=31

Peripherals can share the same ID, which has 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)

8.1.3 Peripherals with shared ID

Peripherals sharing ID [1...n] and a 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, perform the following 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 (INTENCLR = 0xFFFFFFFF).
4. Configure the peripheral being enabled. Do not rely on the inherited configuration from the disabled peripheral.
5. Enable the peripheral.

For a list of peripherals that share an ID, see Instantiation on page 141.

8.1.4 Peripheral registers

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

PSEL registers must be set before a peripheral is enabled or started. Updating PSEL registers while the peripheral is running can cause undefined behavior. 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.

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

Most of the register values are not retained 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 88.

8.1.5 Bit set and clear

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

This bit 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. 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.

8.1.6 Tasks

Tasks 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 138.

8.1.7 Events

Events notify peripherals and the CPU about events that have happened, such as 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 toggles the corresponding event signal and updates the event register to show an event has been generated, see figure Peripheral interface on page 138. An event register is cleared when firmware writes a `0` to that register. A peripheral can continually generate events when the event register is `1`.

8.1.8 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 138.

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 94 for details.

8.1.9 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 the same as connecting a task and event outside the peripheral through the DPPI. The propagation delay for a shortcut 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.

8.1.10 Interrupts

All peripherals support interrupts generated by events.

A peripheral can occupy single or multiple interrupts. For single interrupts, 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). In this case, only single INTEN registers are available.

Events generated by a peripheral can be configured to generate interrupts using registers INTEN, INTENSET, and INTENCLR. Multiple events can be enabled to generate interrupts simultaneously. Event registers in the peripheral register event group indicate the source.

Some peripherals implement only INTENSET and INTENCLR registers. 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 INTEN.

The INTPEND register contains the interrupt pending status of events generated by a peripheral. This is a read-only register.

Peripherals implementing multiple interrupts have several INTEN registers that follow the convention of INTENn, where n is the interrupt number from the peripheral. This also applies to corresponding INTPEND, INTENSET, and INTENCLR registers. This feature enables any event to generate an interrupt from the peripheral.

Peripherals implementing more than 32 events have access to multiple INTEN registers that follow the convention of INTENn, where n is the event group number. The 32 lowest events in the peripheral make event group 0. The next 32 events in the peripheral make event group 1, and so on. This convention is also applicable for corresponding INTPEND, INTENSET, and INTENCLR registers.

Peripherals implementing both multiple interrupts and more than 32 events have multiple INTEN registers. In this case, registers follow the convention of INTENnm, where n is interrupt number from the peripheral and m is event group number. This convention is also applicable for corresponding INTPEND, INTENSET, and INTENCLR registers.

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

To ensure the lowest possible power consumption while in sleep, perform either of the following steps on any pending interrupts:

- Clear the pending interrupt by writing `0` to the corresponding EVENT register
- Disable the interrupt by using the INTEN or INTENCLR registers

This has to be done even if the peripheral is disabled in its ENABLE or POWER register.

The relationship between tasks, events, shortcuts, and interrupts is illustrated in Peripheral interface on page 138.

8.1.10.1 Interrupt clearing and disabling

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

Interrupts are immediately re-triggered until cleared. Routines for software interrupt services continue to execute, even if a new event has not been received.

8.2 Instantiation

ID	Base address	Instance	Description
66	0x40042000	DPPIC00	DPPI controller DPPIC00
68	0x40044000	PPIB00	PPI bridge PPIB00

ID	Base address	Instance	Description
69	0x40045000	PPIB01	PPI bridge PPIB01
74	0x4004A000	AAR00	Accelerated address resolver 00
74	0x4004A000	CCM00	AES CCM mode encryption CCM00, running of HCLKCORE
75	0x4004B000	ECB00	AES ECB mode encryption 00
78	0x4004E000	RRAMC	RRAM Non-Volatile Memory Controller
82	0x40052000	CTRLAP	Control access port CPU side
83	0x40053000	TAD	Trace and debug control
85	0x40055000	TIMER00	Timer TIMER00
88	0x40058000	EGU00	Event generator unit EGU00
89	0x40059000	CRACEN	True Random Number Generator
89	0x40059000	CRACENCORE	True Random Number Generator
130	0x40082000	DPPIC10	DPPI controller DPPIC10
131	0x40083000	PPIB10	PPI bridge PPIB10
132	0x40084000	PPIB11	PPI bridge PPIB11
133	0x40085000	TIMER10	Timer TIMER10
135	0x40087000	EGU10	Event generator unit EGU10
138	0x4008A000	RADIO	2.4 GHz radio RADIO
194	0x400C2000	DPPIC20	DPPI controller DPPIC20
195	0x400C3000	PPIB20	PPI bridge PPIB20
196	0x400C4000	PPIB21	PPI bridge PPIB21
197	0x400C5000	PPIB22	PPI bridge PPIB22
198	0x400C6000	SPIM20	SPI controller SPIM20
198	0x400C6000	SPIS20	SPI peripheral SPIS20
198	0x400C6000	TWIM20	Two-wire interface controller TWIM20
198	0x400C6000	TWIS20	Two-wire interface target TWIS20
198	0x400C6000	UARTE20	Universal asynchronous receiver/transmitter UARTE20
199	0x400C7000	SPIM21	SPI controller SPIM21
199	0x400C7000	SPIS21	SPI peripheral SPIS21
199	0x400C7000	TWIM21	Two-wire interface controller TWIM21
199	0x400C7000	TWIS21	Two-wire interface target TWIS21
199	0x400C7000	UARTE21	Universal asynchronous receiver/transmitter UARTE21
200	0x400C8000	SPIM22	SPI controller SPIM22
200	0x400C8000	SPIS22	SPI peripheral SPIS22
200	0x400C8000	TWIM22	Two-wire interface controller TWIM22
200	0x400C8000	TWIS22	Two-wire interface target TWIS22
200	0x400C8000	UARTE22	Universal asynchronous receiver/transmitter UARTE22
201	0x400C9000	EGU20	Event generator unit EGU20
202	0x400CA000	TIMER20	Timer TIMER20
207	0x400CF000	MEMCONF	Memory Configuration MEMCONF
210	0x400D2000	PWM20	Pulse width modulation unit PWM20
213	0x400D5000	SAADC	Successive approximation analog-to-digital converter SAADC
215	0x400D7000	TEMP	Temperature sensor TEMP
216	0x400D8200	P1	General purpose input and output, port P1
218	0x400DA000	GPIOTE20	8 channels and 1 interrupts for GPIO port P1 GPIO tasks and events GPIOTE20
224	0x400E0000	QDEC20	Quadrature decoder QDEC20
226	0x400E2000	GRTC	Global RTC GRTC
239	0x400EF000	TAMPC	Tamper controller TAMPC
258	0x40102000	DPPIC30	DPPI controller DPPIC30
259	0x40103000	PPIB30	PPI bridge PPIB30
264	0x40108000	WDT30	Watchdog timer WDT30
266	0x4010A000	P0	General purpose input and output, port P0

ID	Base address	Instance	Description
268	0x4010C000	GPIOTE30	4 channels and 1 interrupts for GPIO port P0 GPIO tasks and events GPIOTE30
270	0x4010E000	CLOCK	Clock control
270	0x4010E000	POWER	Power control
270	0x4010E000	RESET	Reset status
288	0x40120000	OSCILLATORS	Oscillator control
288	0x40120000	REGULATORS	Regulator control
N/A	0x00FFC000	FICR	Factory information configuration
N/A	0x00FFD000	UICR	User information configuration

Table 25: Instantiation table

8.3 AAR — Accelerated address resolver

Accelerated address resolver is a cryptographic support function for implementing the Resolvable Private Address Resolution Procedure described in the *Bluetooth Core specification*.

The main features of AAR are:

- Memory-to-memory operations using Scatter/Gather DMA
- Real-time address resolution on incoming packets
- Multiple IRK resolution

The procedure allows two devices that share a secret key to generate and resolve a hash based on their device address. AAR 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*).

The inputs and outputs of AAR are illustrated in the following figure.

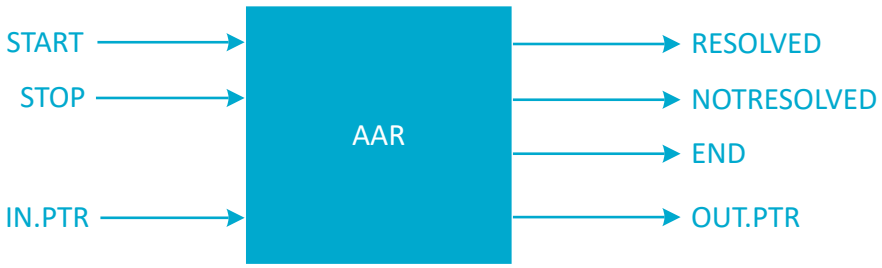


Figure 33: AAR block diagram

8.3.1 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, ECB operation will be aborted and an ERRORECB event will be generated by ECB.

Additionally, AAR shares registers and other resources with the peripherals that have the same ID as AAR. See Peripherals with shared ID on page 139 for more information.

8.3.2 Resolving a resolvable address

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

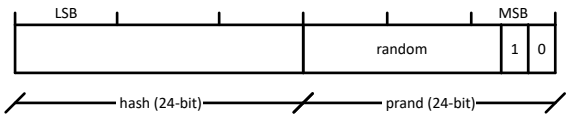


Figure 34: Resolvable address

To resolve an address, IN.PTR must point to a job list describing both the Hash and Prand parts of the private resolvable address (DEVICEADDR) field from the Bluetooth packet, as well as a number of Identity Resolving Keys (IRK). This is illustrated in the examples below. How many IRKs are used is determined by the number of IRKs in the job list. See EasyDMA on page 146 for an introduction to EasyDMA job lists.

The resolver is started by triggering the START task. A RESOLVED event is generated if AAR manages to resolve the address using one of the Identity Resolving Keys (IRK). AAR will generate a NOTRESOLVED event if it is not able to resolve the address using the specified list of IRKs. If there are no IRKs in the joblist, the NOTRESOLVED event is generated.

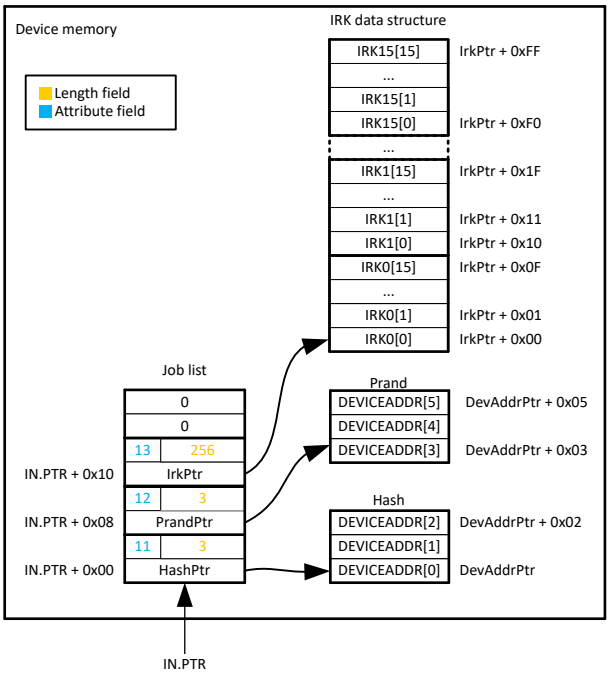


Figure 35: Address resolution with 16 IRKs and DEVICEADDR preloaded into RAM

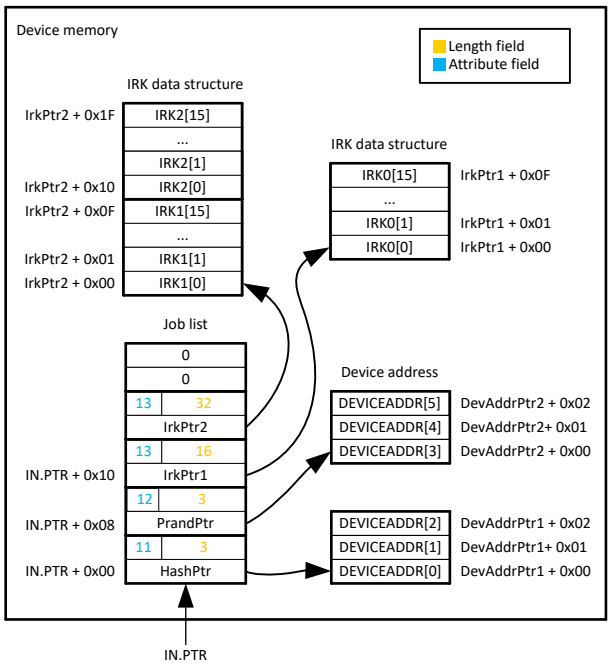


Figure 36: Address resolution with packet device address preloaded into multiple RAM locations, and three IRK keys

AAR will go through the list of available IRKs in the job list, and for each IRK try to resolve the address according to the Resolvable Private Address Resolution Procedure described in the *Bluetooth* Specification[1]. The time it takes to resolve an address may vary depending on where in the list the resolvable address is located. The resolution time will also be affected by RAM accesses performed by other peripherals and the CPU. See the Electrical specifications for more information about resolution time.

Note: Maximum number of IRKs supported in a job list is 4096.

AAR only compares the received address to those programmed in the module without checking the address type.

AAR will stop when it has managed to resolve the maximum number allowed, specified in the MAXRESOLVED register. Each time AAR resolves an IRK, the index of the corresponding IRK is written to memory through the output job list in OUT.PTR. For each IRK found, OUT.AMOUNT is updated accordingly.

The output job list must define a memory region large enough to hold list of resolved IRK indices. This is calculated from the MAXRESOLVED register, where each IRK index occupies two bytes in memory. In the example below, the `n` indicates the number of bytes (size) in the resolved IRK index array. The value of `n` must be exactly 2 times the value of MAXRESOLVED.

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

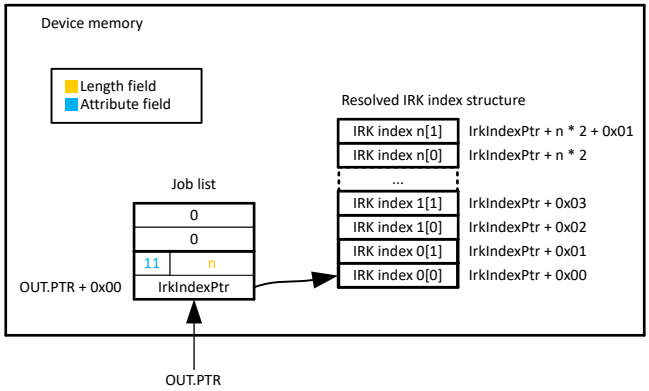


Figure 37: Resolved IRK index structure at RAM

At the end of the operation, AAR will generate the END event.

Triggering the STOP task will stop AAR. If AAR is stopped before the operation has completed, the END, RESOLVED, and NOTRESOLVED events are not generated. However, if STOP is triggered close to the end of the operation the events can be generated.

8.3.3 EasyDMA

This peripheral implements EasyDMA with scatter-gather functionality for reading from memory without CPU involvement.

The scatter-gather functionality allows EasyDMA to collect data from multiple memory regions, instead of one contiguous block. The memory regions are described by a job list, called input job list. The job list consists of one or more job entries that consist of a 32-bit address field, 8-bit attribute field, and 24-bit length field. A job list ends with a zero filled job entry.

The input job list must have separate entries for the following entries:

1. The three first bytes of the resolvable private address (the 24-bit hash)
2. The three following bytes of the resolvable private address (the 24-bit prand)
3. The IRKs

The attribute field of each of these entries identify the input job and must be set according to the following table.

Attribute	Value
Hash	11
Prand	12
Irk	13

Table 26: Attribute field for input job list

If the IN.PTR register or the entries in the input job list are not pointing to memory connected to the DMA bus, an EasyDMA transfer may result in a HardFault or memory corruption. See Memory on page 11 for more information about the different memory regions and DMA connectivity.

The EasyDMA will have finished accessing the RAM when the END, RESOLVED, or NOTRESOLVED events are generated.

For instances supporting DMA error detection, the ERRORSTATUS register will report if a bus error has occurred during DMA access. To see if DMA error detection is supported, see the the instance's configuration in Instantiation on page 141.

The list of the resolved IRK indices are stored in memory using the output job list, configured by OUT.PTR.

Attribute	Value
Resolved Irk index	11

Table 27: Attribute field for output job list

Example

The figure below shows an example of a job list with three job entries. Each of the entries point to a memory address, and the length field describes how many bytes of data is stored at that address. There are three blocks of memory in use

- Hash, an array of length 3
- Prand, an array of length 3
- Irk, an array of at least length 16

The data pointed to from the job list is what is fed into the module and processed according to the peripheral's operation. The entries of the job list comprises pointers to the individual arrays, as well as their sizes. Job entries with length greater than one are processed in little endian order.

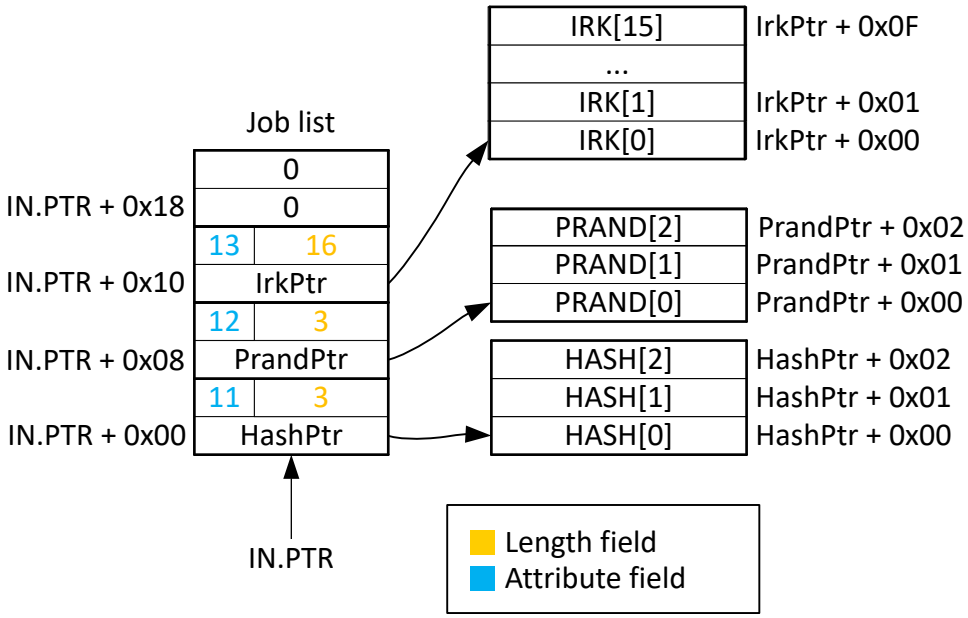


Figure 38: EasyDMA Scatter-Gather job list example

8.3.4 Use case example for chaining 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 Hash and Prand part of the job list must point to the part of the packet containing the device address.

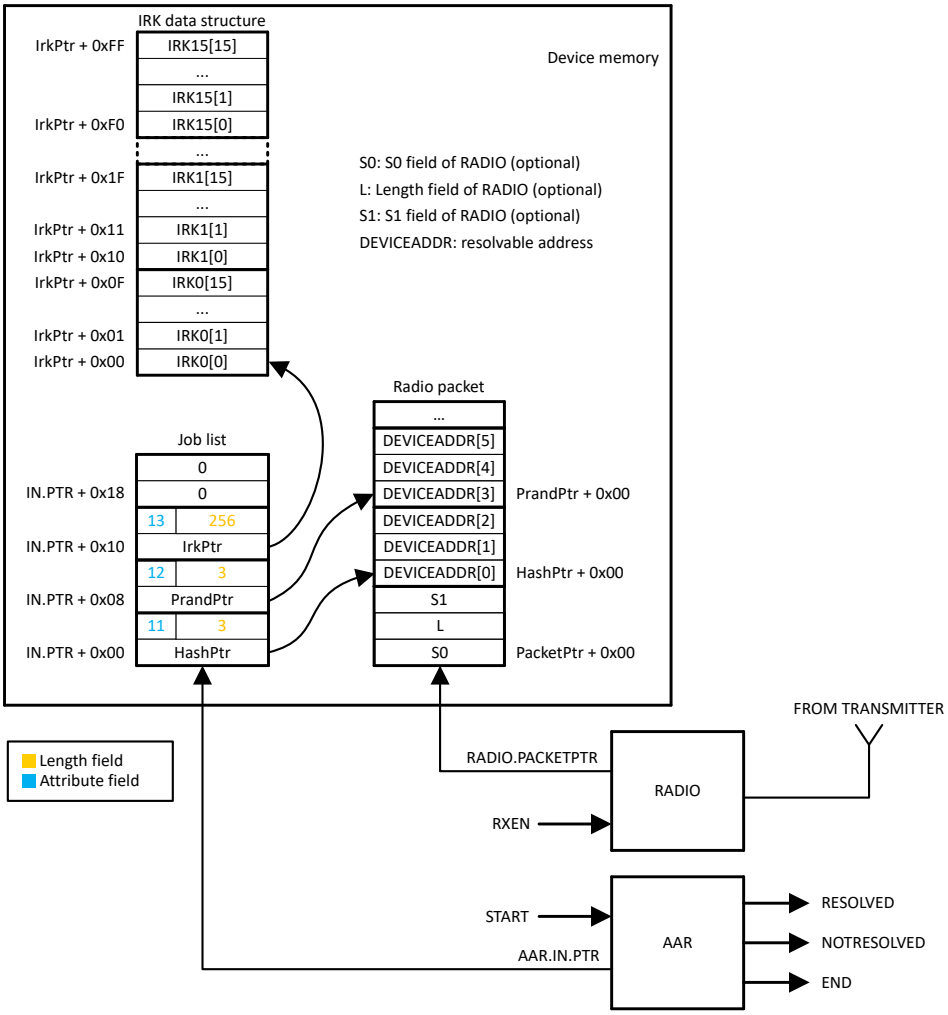


Figure 39: Address resolution with packet loaded into RAM by RADIO

8.3.5 Registers

Instances

Instance	Domain	Base address	Description
AAR00	GLOBAL	0x4004A000	Accelerated address resolver 00

Register overview

Register	Offset	Description
TASKS_START	0x000	Start resolving addresses based on IRKs specified in the IRK data structure
TASKS_STOP	0x004	Stop resolving addresses
SUBSCRIBE_START	0x080	Subscribe configuration for task START
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
EVENTS_END	0x100	Address resolution procedure complete or ended due to an error
EVENTS_RESOLVED	0x104	Address resolved
EVENTS_NOTRESOLVED	0x108	Address not resolved
EVENTS_ERROR	0x10C	Operation aborted because of a STOP task or due to an error
PUBLISH_END	0x180	Publish configuration for event END
PUBLISH_RESOLVED	0x184	Publish configuration for event RESOLVED

Register	Offset	Description
PUBLISH_NOTRESOLVED	0x188	Publish configuration for event NOTRESOLVED
PUBLISH_ERROR	0x18C	Publish configuration for event ERROR
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSTATUS	0x404	Error status
ENABLE	0x500	Enable AAR
MAXRESOLVED	0x508	Maximum number of IRKs to resolve
IN.PTR	0x530	Input pointer
OUT.PTR	0x538	Output pointer
OUT.AMOUNT	0x53C	Number of bytes transferred in the last transaction

8.3.5.1 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

8.3.5.2 TASKS_STOP

Address offset: 0x004

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

8.3.5.3 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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.3.5.4 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.3.5.5 EVENTS_END

Address offset: 0x100

Address resolution procedure complete or ended 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_END			Address resolution procedure complete or ended due to an error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.3.5.6 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

8.3.5.7 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

8.3.5.8 EVENTS_ERROR

Address offset: 0x10C

Operation aborted because of a STOP 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_ERROR			Operation aborted because of a STOP task or due to an error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.3.5.9 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.3.5.10 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event RESOLVED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.3.5.11 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		[0..255]	DPPI channel that event NOTRESOLVED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.3.5.12 PUBLISH_ERROR

Address offset: 0x18C

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.3.5.13 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 W1S	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	RESOLVED			Write '1' to enable interrupt for event RESOLVED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	NOTRESOLVED			Write '1' to enable interrupt for event NOTRESOLVED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	ERROR			Write '1' to enable interrupt for event ERROR
			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				D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

8.3.5.14 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 W1C	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	RESOLVED			Write '1' to disable interrupt for event RESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	NOTRESOLVED			Write '1' to disable interrupt for event NOTRESOLVED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.3.5.15 ERRORSTATUS

Address offset: 0x404

Error 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	ERRORSTATUS			Error status when the ERROR event is generated
			NoError	0	No errors have occurred
			PrematureInptrEnd	1	End of INPTR job list before data structure was read.
			PrematureOutptrEnd	2	End of OUTPTR job list before data structure was read.
			DmaError	4	Bus error during DMA access.

8.3.5.16 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																															
A	RW	ENABLE																																	
			Disabled	0																															
			Enabled	3																															

8.3.5.17 MAXRESOLVED

Address offset: 0x508

Maximum number of IRKs to resolve

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	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 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	RW	MAXRESOLVED		1..4095																															
				After MAXRESOLVED number of IRKs have been resolved, AAR will stop processing and generate the END event																															

8.3.5.18 IN

IN EasyDMA channel

8.3.5.18.1 IN.PTR

Address offset: 0x530

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																															
A	RW	PTR																																	

8.3.5.19 OUT

OUT EasyDMA channel

8.3.5.19.1 OUT.PTR

Address offset: 0x538

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	PTR			Output pointer

8.3.5.19.2 OUT.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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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..255]	Number of bytes written to memory after triggering the START task. Each resolved IRK index uses two bytes.

8.4 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 (encryption/decryption) during data transfer.

The main features of CCM are:

- Memory-to-memory packet encryption and decryption operations using Scatter/Gather DMA
- Support for Bluetooth requirements and algorithm as defined in IETF RFC3610
- Support for IEEE 802.15.4
- Concurrent operation with RADIO

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.



Figure 40: CCM Overview

CCM generates an encrypted keystream that is applied to input data using the XOR operation and generates an M byte MAC 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 the radio. All operations can complete within the packet RX or TX time. CCM on this device is implemented to support the 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.

CCM uses EasyDMA to read/write additional authenticated data, plain text and cipher text.

Two operations are supported:

- Packet encryption
- Packet decryption

All operations are done in compliance with the *Bluetooth Core Specification*, as well as IEEE 802.15.4.

8.4.1 Shared resources

The CCM shares the same AES module as the AAR and ECB peripherals. The ECB will always have the lowest priority. If an operation is aborted due to a conflict among the shared resources, an ERROR event will be generated.

Additionally, the CCM shares registers and other resources with the peripherals that have the same ID as the CCM. See Peripherals with shared ID on page 139 for more information.

8.4.2 Encryption and decryption

CCM supports both packet encryption and decryption.

The following table shows the different CCM input/output and parameters supported by the CCM module for encryption and decryption:

Parameter	Valid input	Description
M	0, 4, 6, 8, 10, 12, 14, 16	Number of bytes in the authentication field
L	2 (fixed)	Number of bytes in the length field
l(a)	0-65279	Number of bytes in additional authenticated data
l(m)	0-(65535 - M)	Number of bytes in the message to authenticate and encrypt
l(c)	0-65535	Number of bytes in the encrypted message; l(m) + M bytes
a	l(a) number of bytes	Additional authenticated data
m	l(m) number of bytes	Message to authenticate and encrypt
c	l(c) number of bytes	Encrypted message

Table 28: CCM Parameters

In addition to the parameters listed above, the CCM requires two sets of data: a 128-bit key and a 128-bit nonce. These are supplied via dedicated register interfaces: KEY.VALUE registers for the 128-bit key, and NONCE.VALUE registers for the 128-bit nonce. The 128-bit key in the KEY.VALUE registers is stored in reverse byte order relative to the payload. For example, using the sample session key from the Bluetooth Core Specification v5.4, Volume 6, Part C, chapter 1.2:

- Session Key (SK): 99AD1B5226A37E3E058E3B8E27C2C666

The KEY.VALUE registers are populated as follows:

- KEY.VALUE[0] = `0x27C2C666`
- KEY.VALUE[1] = `0x058E3B8E`
- KEY.VALUE[2] = `0x26A37E3E`
- KEY.VALUE[3] = `0x99AD1B52`

The same reverse byte order is used for the NONCE.VALUE registers. For the packet example "3. Data packet1" with the following values:

- IV: DEAFBABEBADCAB24
- Direction Bit: 1
- Packet Counter: 1

The NONCE.VALUE registers are populated as follows:

- NONCE.VALUE[0] = `0xBEBAAFDE`
- NONCE.VALUE[1] = `0x24ABDCBA`
- NONCE.VALUE[2] = `0x00000080`
- NONCE.VALUE[3] = `0x00000001`

Note: Although the NONCE in the example above is 13 bytes, it must be written as a 16-byte value with the first 3 bytes zero-padded.

Note: The KEY and NONCE byte order is reversed compared to the NRF52 and NRF53 series devices.

8.4.2.1 Encryption

During packet encryption, CCM will read the unencrypted packet located in memory at the address specified in register IN.PTR, encrypt the packet and append an M byte long message authentication code (MAC) field to the packet.

The message to authenticate and encrypt (m) and additional authenticated data (a) are included in the MAC generation. The first byte in the packet header can be masked by configuring the ADATAMASK register. This is useful for Bluetooth header masking. For protocols other than Bluetooth, the ADATAMASK register must be set to 0xFF for correct CCM operation; the reset value is configured to support Bluetooth.

Encryption is started by triggering the START task with the MODE register set to `Encryption`. The END event will be generated when packet encryption is completed.

The AES CCM will modify the l(c) output field of the packet to adjust for the appended MAC field, that is, add MODE.MACLEN bytes to l(m), and store the resulting packet back into memory at the locations specified in the OUT.PTR list, as illustrated in the following figure. The maximum length of l(m) plus MODE.MACLEN cannot exceed 65535 bytes.

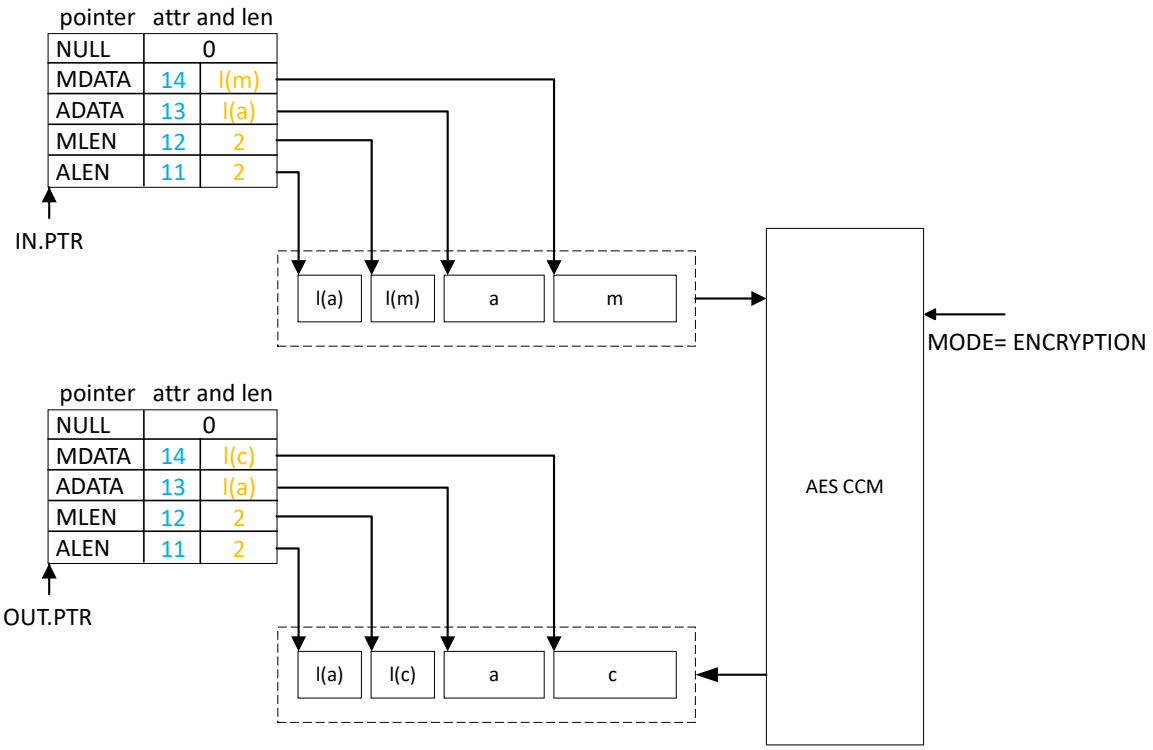


Figure 41: Encryption

If the following occurs, the ERROR event is generated, the CCM stops, and the ERRORSTATUS register will report the type of error that triggered the ERROR event:

- The IN.PTR job list ends before reading out the complete CCM data structure
- The OUT.PTR job list ends before writing out the complete encrypted CCM data structure
- The CCM is not able to operate fast enough to run concurrently with the RADIO as the RADIO transmits the encrypted packet.
- The EasyDMA engine encounters an error, see EasyDMA and ERROR event on page 162

Any values of l(m) and l(a) are allowed. If encrypting empty packets, i.e. l(m) = l(a) = 0, no encryption will take place; the END event is generated, and CCM operation is stopped.

For Bluetooth (MODE.PROTOCOL=`Ble`), valid packets with 0 payload (l(a) is larger than 0 but l(m) is 0) will not be authenticated but instead moved unmodified through the AES CCM peripheral, and thus no MAC will be generated.

For IEEE 802.15.4 (MODE.PROTOCOL=`Ieee802154`), valid packets with 0 payload (l(a) is larger than 0 but l(m) is 0) will be authenticated, and thus a MAC will be generated as part of the output data.

8.4.2.2 Decryption

During packet decryption, CCM will read the encrypted packet located in memory at the address specified in the IN.PTR pointer, decrypt the packet, authenticate the packet's MAC field and generate the appropriate MAC status.

The encrypted message in (c), is decrypted and authenticated together with additional authenticated data (a) and then matched against the decrypted MAC value. The decrypted MAC value is part of (c). Bits in the first byte of the data can be masked away before calculating the MAC value by configuring the ADATAMASK register. This is useful for Bluetooth header masking. For protocols other than Bluetooth, the ADATAMASK register must be set to 0xFF for correct CCM operation; the reset value is configured to support Bluetooth.

Decryption is started by triggering the START task with the MODE register set to `FastDecryption`.

CCM will write the l(m) value of the decrypted packet to the location provided in OUT.PTR, and then store the decrypted packet into memory at the locations given by the OUT.PTR list as illustrated in the following figure.

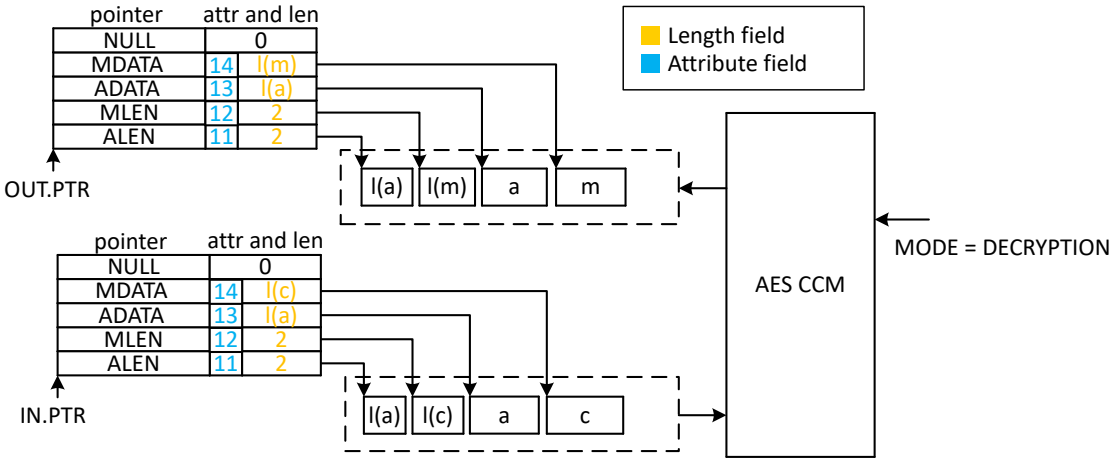


Figure 42: Decryption

For Bluetooth (MODE.PROTOCOL=`Ble`), CCM is only able to authenticate messages where l(c) is at least MACLEN+1 bytes long. If l(c) is less than MACLEN+1, CCM will generate an END event and clear the MACSTATUS (indicating MAC check failure). Furthermore, empty packets (l(c)=0) will be moved unmodified through the AES CCM peripheral even though ERROR event shall be generated. In any other case that leads to a failed MACSTATUS or an ERROR event, the contents of the job addresses given in OUT.PTR are undefined.

For IEEE 802.15.4 (MODE.PROTOCOL=`Ieee802154`), CCM will also perform authentication on messages where only ADATA is present (i.e. l(m)=0 and l(a)>0). In this case MACSTATUS reflects the result of the authentication. If l(c)<MACLEN, then the ERROR event is generated, and the contents of the locations given in OUT.PTR are undefined.

If the following occurs, the ERROR event is generated, and CCM is stopped.

- The IN.PTR job list ends before reading out the complete CCM data structure
- The OUT.PTR job list ends before writing out the complete decrypted CCM data structure
- The EasyDMA engine encounters an error, see EasyDMA and ERROR event on page 162

If the IN.PTR or OUT.PTR job lists do not end before the complete encrypted/decrypted CCM data structures are read, the END event is generated and CCM operation is stopped.

8.4.3 Encrypting packets in radio transmit mode

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

The OUT.PTR pointer in the AES CCM must therefore point to the same memory location as the PACKETPTR pointer in the radio, see Example configuration of encryption during radio transmission on page 160.

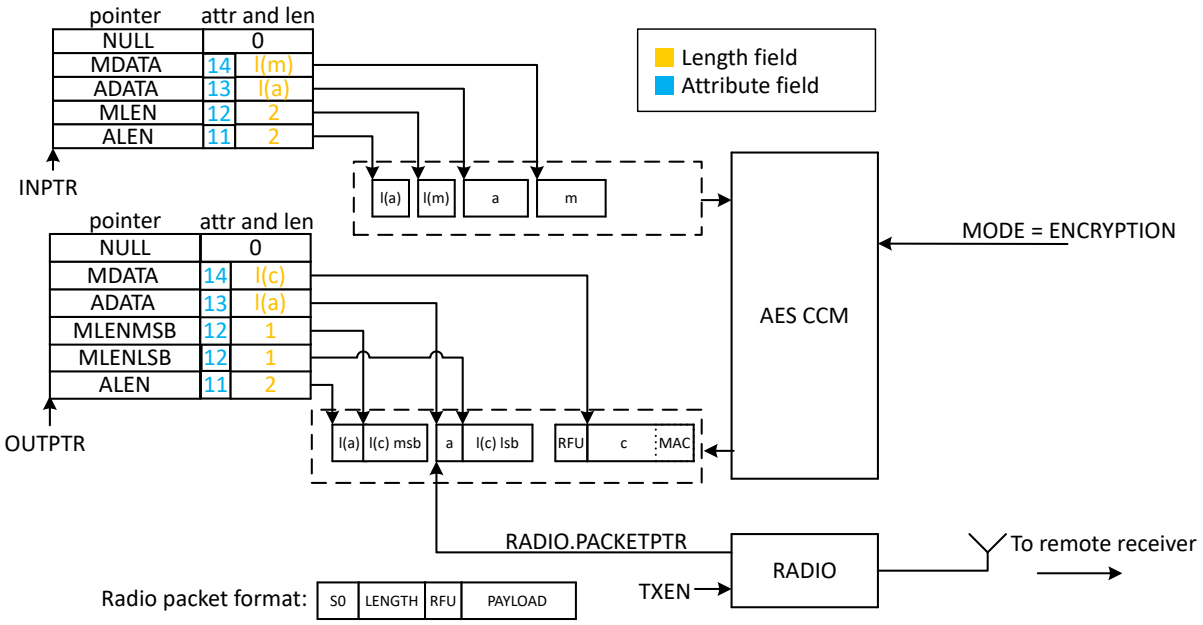


Figure 43: Example configuration of encryption during radio transmission

The START task must be triggered by RADIO READY event to ensure that the payload is encrypted in time for radio transmission. This is illustrated in the following figure, using a PPI connection between RADIO.EVENTS_READY and CCM.TASKS_START.

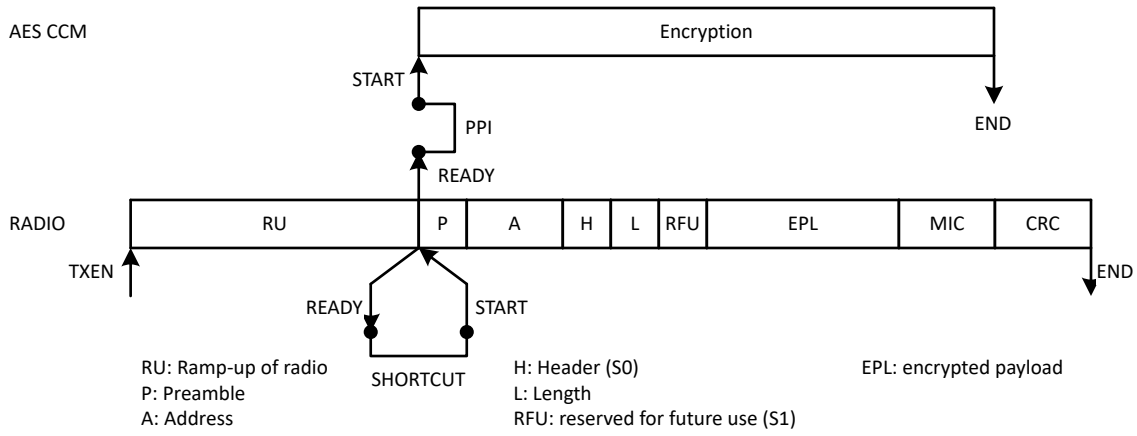


Figure 44: Radio transmission with encryption using a PPI connection

8.4.4 Decrypting packets received by the radio

To decrypt a packet received by the radio immediately upon its reception, CCM can be started when the RADIO PAYLOAD event is generated. The packet is decrypted when the CCM.END event is generated. Typically, CCM will decrypt the packet before or during the reception of the CRC. However, if the packet is large and the bitrate is high, CCM will not finish before the PHYEND event, but shortly afterward. After the CCM.END event is generated; the MACSTATUS can be checked.

AES CCM must therefore operate on the same memory location as RADIO, as illustrated in the following figure.

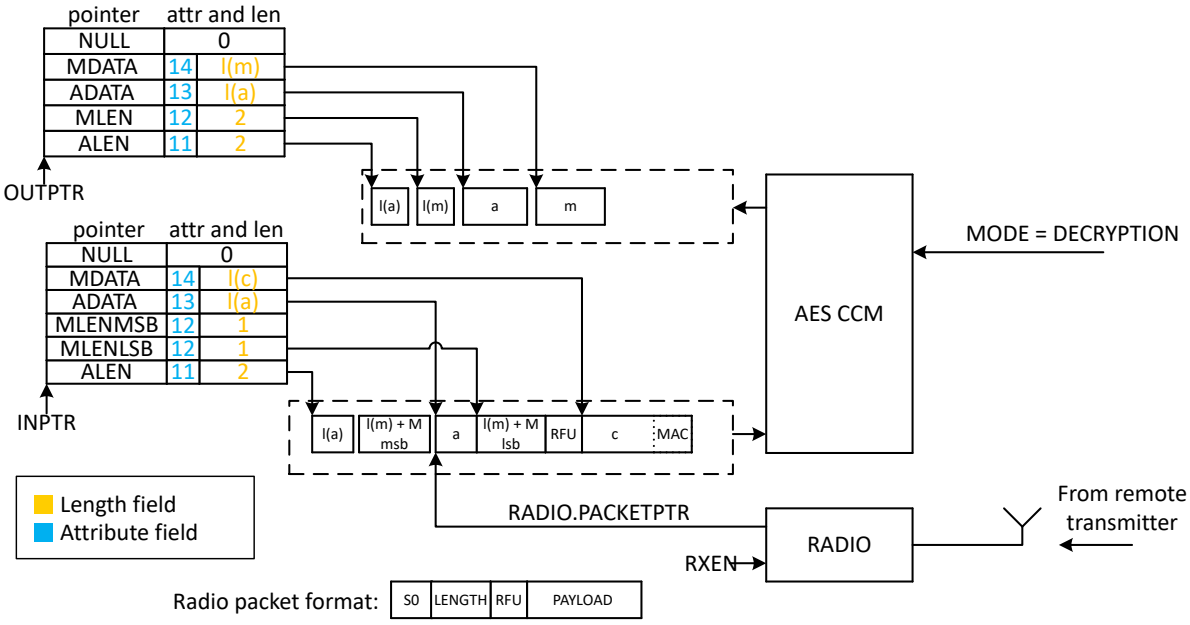


Figure 45: Example configuration of CCM for decrypting a packet as it is received by the RADIO

8.4.5 CCM data structure

The input and output data structures are located in memory specified by IN.PTR and OUT.PTR on page 169 respectively.

Both IN.PTR and OUT.PTR point to a scatter/gather job list. This job list must contain all the fields listed in the attribute field table. Each job list must be terminated with a 0 filled job entry. If either of the IN.PTR or OUT.PTR job list is not terminated, then the behavior of CCM is undefined.

The job list consists of one or more job entries each containing a 32-bit address field, an 8-bit attribute field, and a 24-bit length field. A job list ends with a zero-filled job entry. The EasyDMA job list example below illustrates a job list that points to three different memory sections with varying lengths. The data pointed to by the job list is fed into the module to be processed according to the CCM operation. Job entries with a length greater than one byte are processed in little endian order.

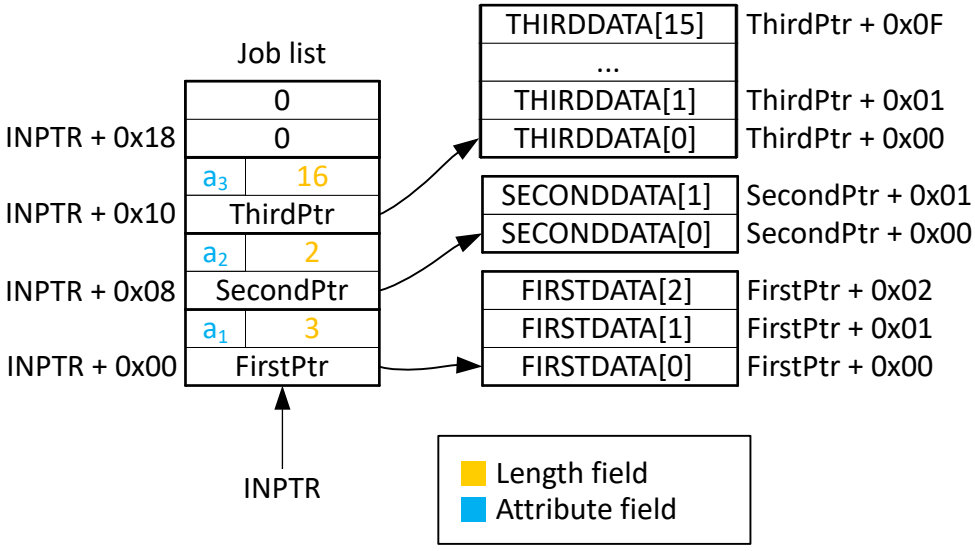


Figure 46: EasyDMA job list example

The attribute field identifies the job and must be set according to the following table.

Attribute	Value
Alen	11
Mlen	12
Adata	13
Mdata	14

Table 29: Attribute field

8.4.6 EasyDMA and ERROR event

The CCM implements an EasyDMA with scatter/gather mechanism for reading and writing to memory.

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

EasyDMA will have finished accessing the memory when the END event is generated.

If the IN.PTR and the OUT.PTR are not pointing to memory with DMA connectivity, an EasyDMA transfer may result in a HardFault or memory corruption. See Memory on page 11 for more information about the different memory regions.

For instances supporting DMA error detection, the ERRORSTATUS register will report if a bus error has occurred during DMA access.

8.4.7 Registers

Instances

Instance	Domain	Base address	Description
CCM00	GLOBAL	0x4004A000	AES CCM mode encryption CCM00, running of HCLKCORE

Configuration

Instance	Domain	Configuration
CCM00	GLOBAL	Does not support on-the-fly decryption. No bypass mode available in the MODE register.

Register overview

Register	Offset	Description
TASKS_START	0x000	Start encryption/decryption. This operation will stop by itself when completed.
TASKS_STOP	0x004	Stop encryption/decryption
TASKS_RATEOVERRIDE	0x008	Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption
SUBSCRIBE_START	0x080	Subscribe configuration for task START
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
SUBSCRIBE_RATEOVERRIDE	0x088	Subscribe configuration for task RATEOVERRIDE
EVENTS_END	0x104	Encrypt/decrypt complete or ended because of an error
EVENTS_ERROR	0x108	CCM error event

Register	Offset	Description
PUBLISH_END	0x184	Publish configuration for event END
PUBLISH_ERROR	0x188	Publish configuration for event ERROR
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
MACSTATUS	0x400	MAC check result
ERRORSTATUS	0x404	Error status
ENABLE	0x500	Enable
MODE	0x504	Operation mode
KEY.VALUE[n]	0x510	128-bit AES key
NONCE.VALUE[n]	0x520	13-byte NONCE vector Only the lower 13 bytes are used
IN.PTR	0x530	Input pointer Points to a job list containing unencrypted CCM data structure in Encryption mode Points to a job list containing encrypted CCM data structure in Decryption mode
OUT.PTR	0x538	Output pointer Points to a job list containing encrypted CCM data structure in Encryption mode Points to a job list containing decrypted CCM data structure in Decryption mode
RATEOVERRIDE	0x544	Data rate override setting.
ADATAMASK	0x548	CCM adata mask.

8.4.7.1 TASKS_START

Address offset: 0x000

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_START			Start encryption/decryption. This operation will stop by itself when completed.
			Trigger	1	Trigger task

8.4.7.2 TASKS_STOP

Address offset: 0x004

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

8.4.7.3 TASKS_RATEOVERRIDE

Address offset: 0x008

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**

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

8.4.7.4 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.4.7.5 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.4.7.6 SUBSCRIBE_RATEOVERRIDE

Address offset: 0x088

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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RATEOVERRIDE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.4.7.7 EVENTS_END

Address offset: 0x104

Encrypt/decrypt complete or ended because of 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_END			Encrypt/decrypt complete or ended because of an error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.4.7.8 EVENTS_ERROR

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
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.4.7.9 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.4.7.10 PUBLISH_ERROR

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.4.7.11 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
	W1S				
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to enable interrupt for event ERROR
	W1S				
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.4.7.12 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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
	W1C				
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to disable interrupt for event ERROR
	W1C				
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.4.7.13 MACSTATUS

Address offset: 0x400

MAC 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																															
A	R	MACSTATUS																																	
			CheckFailed	0																															
			CheckPassed	1																															

8.4.7.14 ERRORSTATUS

Address offset: 0x404

Error 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	R	ERRORSTATUS																																	
			NoError	0																															
			PrematureInptrEnd	1																															
			PrematureOutptrEnd	2																															
			EncryptionTooSlow	3																															
			DmaError	4																															

8.4.7.15 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																															
A	RW	ENABLE																																	
			Disabled	0																															
			Enabled	2																															

8.4.7.16 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									D	D	D					C	C	C						B	B						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	RW	MODE																																	

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D C C C B B 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
			Encryption	0	AES CCM packet encryption mode
			Decryption	1	This mode will run CCM decryption in the speed of the DATARATE field. This enumerator is deprecated.
			FastDecryption	2	AES CCM decryption mode.
				3	
B	RW	PROTOCOL			Protocol and packet format selection
			Ble	0	Bluetooth Low Energy packet format
			Ieee802154	1	802.15.4 packet format
				2	
				3	
C	RW	DATARATE			Radio data rate that the CCM shall run synchronous with
			125Kbit	0	125 Kbps
			250Kbit	1	250 Kbps
			500Kbit	2	500 Kbps
			1Mbit	3	1 Mbps
			2Mbit	4	2 Mbps
			4Mbit	5	4 Mbps
D	RW	MACLEN			CCM MAC length (bytes)
			M0	0	M = 0 This is a special case for CCM* where encryption is required but not authentication
			M4	1	M = 4
			M6	2	M = 6
			M8	3	M = 8
			M10	4	M = 10
			M12	5	M = 12
			M14	6	M = 14
			M16	7	M = 16

8.4.7.17 KEY.VALUE[n] (n=0..3)

Address offset: 0x510 + (n × 0x4)

128-bit 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	VALUE			AES 128-bit key value, bits (32*(i+1))-1 : (32*i)

8.4.7.18 NONCE.VALUE[n] (n=0..3)

Address offset: 0x520 + (n × 0x4)

13-byte NONCE vector

Only the lower 13 bytes are 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	A	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	VALUE			NONCE value, bits (32*(n+1))-1 : (32*n)

8.4.7.19 IN

IN EasyDMA channel

8.4.7.19.1 IN.PTR

Address offset: 0x530

Input pointer

Points to a job list containing unencrypted CCM data structure in Encryption mode

Points to a job list containing encrypted CCM data structure in Decryption 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	PTR			Input pointer

8.4.7.20 OUT

OUT EasyDMA channel

8.4.7.20.1 OUT.PTR

Address offset: 0x538

Output pointer

Points to a job list containing encrypted CCM data structure in Encryption mode

Points to a job list containing decrypted CCM data structure in Decryption 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	PTR			Output pointer

8.4.7.21 RATEOVERRIDE

Address offset: 0x544

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.

Note: The override is only applied when operating in BLE Long Range 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 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	RATEOVERRIDE			Data rate override setting.
			125Kbit	0	125 Kbps
			500Kbit	2	500 Kbps
			1Mbit	3	1 Mbps
			2Mbit	4	2 Mbps
			4Mbit	5	4 Mbps

8.4.7.22 ADATAMASK

Address offset: 0x548

CCM adata mask.

Bitmask for the first adata byte. The masking is done before MAC 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	ADATAMASK			CCM adata mask.

8.5 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).

The main features of ECB are:

- 128-bit AES encryption
- Supports standard AES ECB block encryption
- Memory-to-memory operations using Scatter/Gather DMA

The inputs and outputs of the ECB are illustrated below.

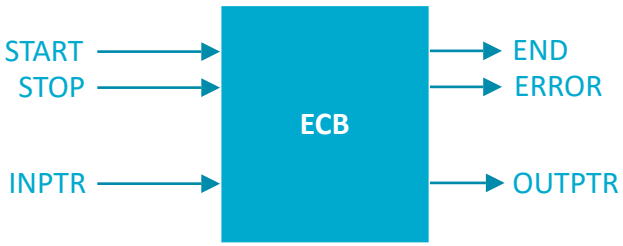


Figure 47: ECB block diagram

AES ECB uses EasyDMA with scatter-gather to access to memory 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 performs a 128 bit AES block encrypt. At the START task, cleartext is loaded into the ECB from memory described by the scatter/gather job list pointed to by INPTR and the ciphertext is written into memory described by the job list pointed to by OUTPTR. When the last cleartext byte has been encrypted and written to OUTPTR, the END event is triggered.

The following figure illustrates how the input and output job lists can be configured. For more details of the joblists, see EasyDMA on page 172.

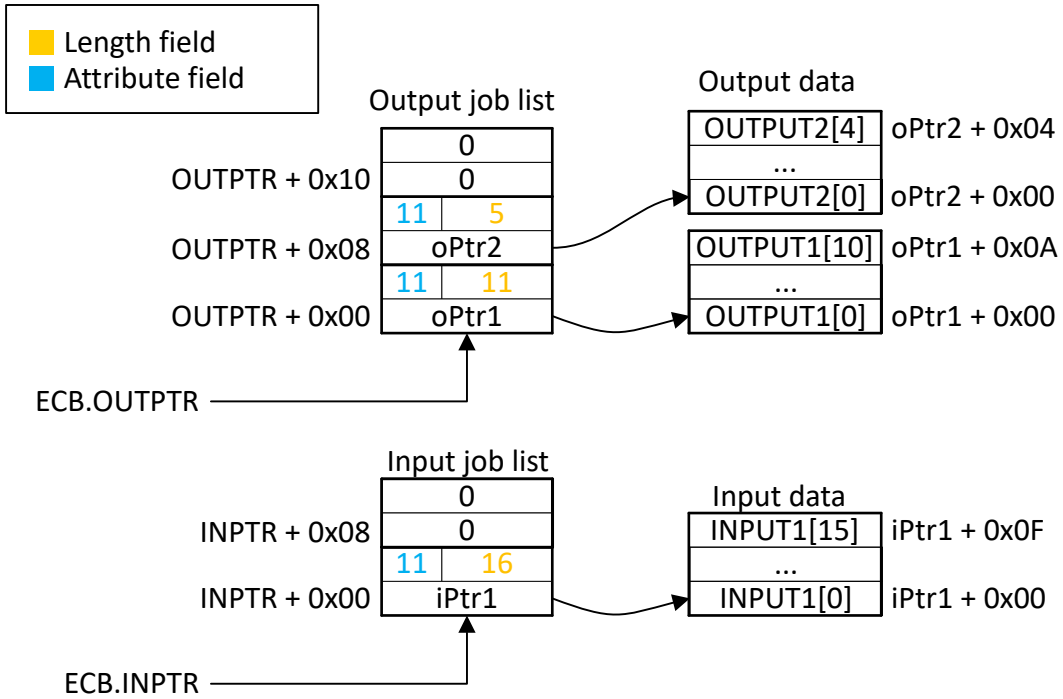


Figure 48: Example job lists for ECB operation

The AES key is set by writing the KEY.VALUE key registers. The same key can be used to encrypt multiple blocks by triggering the START task multiple times.

AES ECB can be stopped by triggering the STOP task.

ECB only supports a single 16-byte block. For different job list sizes the following rules apply:

- If less than 16 bytes is supplied as input, then a PrematureInptrEnd error is triggered. The EVENTS_ERROR will also be set.
- If more than 16 bytes is supplied as input, then only the first 16 bytes are used.
- For an output job, only the number of bytes specified in the job list are copied to memory.

The 128-bit key in the KEY.VALUE registers is stored in reverse byte order relative to the payload. For example, using the sample calculation from the Bluetooth Core Specification v5.4, Volume 6, Part C, chapter 1.1, with the following data:

- Key: 4C68384139F574D836BCF34E9DFB01BF
- Plaintext: 0213243546576879acbdcedfe0f10213
- Expected Encrypted Output: 99ad1b5226a37e3e058e3b8e27c2c666

The KEY.VALUE registers are populated as follows:

- KEY.VALUE[0] = `0x9DFB01BF`
- KEY.VALUE[1] = `0x36BCF34E`
- KEY.VALUE[2] = `0x39F574D8`
- KEY.VALUE[3] = `0x4C683841`

The IN.PTR points to a job that contains the following 16-byte input data array:

```
        {0x02, 0x13, 0x24, 0x35, 0x46, 0x57, 0x68, 0x79, 0xAC, 0xBD, 0xCE, 0xDF, 0xE0,
0xF1, 0x02, 0x13}
```

Once the encryption is complete, the output buffer referenced by the output job will be filled with the following 16-byte array:

```
        {0x99, 0xAD, 0x1B, 0x52, 0x26, 0xA3, 0x7E, 0x3E, 0x05, 0x8E, 0x3B, 0x8E, 0x27,
0xC2, 0xC6, 0x66}
```

Note: The KEY byte order is reversed compared to the NRF52 and NRF53 series devices.

8.5.1 Shared resources

The ECB shares the same AES module as the AAR and CCM peripherals. The ECB will always have lowest priority. If there is a sharing conflict during encryption, the ECB operation will be aborted and an ERROR event will be generated.

8.5.2 EasyDMA

This peripheral implements EasyDMA with scatter-gather functionality for reading from and writing to memory without CPU involvement.

The scatter-gather functionality allows EasyDMA to collect data from multiple memory regions, instead of one contigous block. The memory regions are described by a job list. The job list consists of one or more job entries that consist of a 32-bit address field, 8-bit attribute field, and 24-bit length field. A job list ends with a zero filled job entry. The attribute field must be set to `11`.

If INPTR or OUTPTR pointers or the entries in the job lists are not pointing to memory connected to the DMA bus, an EasyDMA transfer may result in a HardFault or memory corruption. See Memory on page 11 for more information about the different memory regions and DMA connectivity.

The EasyDMA will have finished accessing the RAM when the END or ERROR events are generated.

For instances supporting DMA error detection, the ERRORSTATUS register will report if a bus error has occurred during DMA access. To see if DMA error detection is supported, see the the instance's configuration in Instantiation on page 141.

Example

The figure below shows an example of a job list with three job entries. Each of the entries point to a memory address, and the length field describes how many bytes of data is stored at that address. There are three blocks of memory in use

- FIRSTDATA, an array of length 3
- SECONDDATA, an array of length 2
- THIRDDATA, an array of length 11

The data pointed to from the job list is what is fed into the module and processed according to the peripheral's operation. The entries of the job list comprises pointers to the individual arrays, as well as their sizes. Job entries with length greater than one are processed in little endian order.

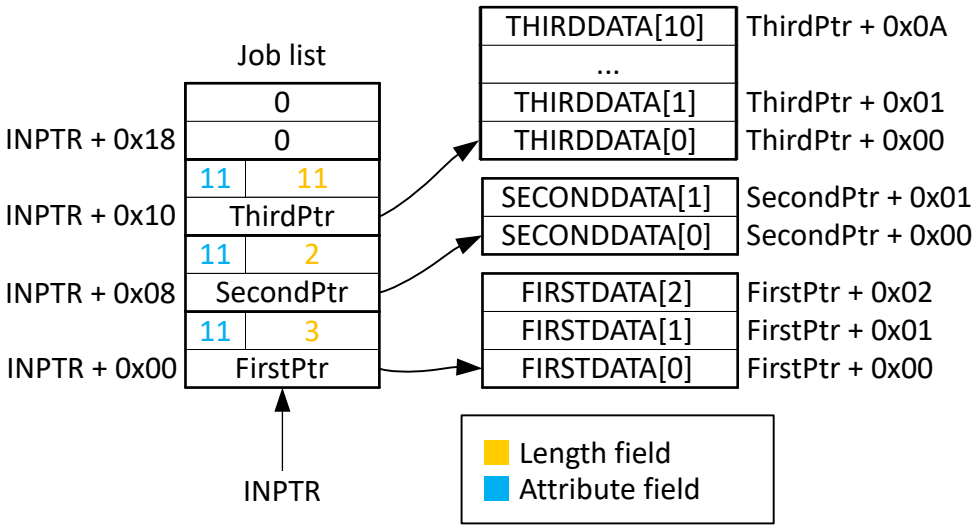


Figure 49: EasyDMA Scatter-Gather job list example

8.5.3 Registers

Instances

Instance	Domain	Base address	Description
ECB00	GLOBAL	0x4004B000	AES ECB mode encryption 00

Configuration

Instance	Domain	Configuration
ECB00	GLOBAL	

Register overview

Register	Offset	Description
TASKS_START	0x000	Start ECB block encrypt
TASKS_STOP	0x004	Abort a possible executing ECB operation
SUBSCRIBE_START	0x080	Subscribe configuration for task START
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
EVENTS_END	0x100	ECB block encrypt complete
EVENTS_ERROR	0x104	ECB block encrypt aborted because of a STOP task or due to an error
PUBLISH_END	0x180	Publish configuration for event END
PUBLISH_ERROR	0x184	Publish configuration for event ERROR
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSTATUS	0x400	Error status
KEY.VALUE[n]	0x510	128-bit AES key
IN.PTR	0x530	Input pointer
OUT.PTR	0x538	Output pointer Points to a job list containing encrypted ECB data structure
CSAA.REFLECTOR	0x53C	Selected Channel Sounding Access Address used in the CS SYNC from Reflector to Initiator
CSAA.INITIATOR	0x540	Selected Channel Sounding Access Address used in the CS SYNC from Initiator to Reflector
CSAA.MODE	0x544	Operation modes

8.5.3.1 TASKS_START

Address offset: 0x000

Start ECB block encrypt

If a crypto operation is already running in the AES core, the START task will not start a new encryption and an ERROR 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_START			Start ECB block encrypt If a crypto operation is already running in the AES core, the START task will not start a new encryption and an ERROR event will be triggered
			Trigger	1	Trigger task

8.5.3.2 TASKS_STOP

Address offset: 0x004

Abort a possible executing ECB operation

If a running ECB operation is aborted by STOP, the ERROR 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	W	TASKS_STOP			Abort a possible executing ECB operation If a running ECB operation is aborted by STOP, the ERROR event is triggered.
			Trigger	1	Trigger task

8.5.3.3 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task START

If a crypto operation is already running in the AES core, the START task will not start a new encryption and an ERROR 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.5.3.4 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

If a running ECB operation is aborted by STOP, the ERROR 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.5.3.5 EVENTS_END

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_END			ECB block encrypt complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.5.3.6 EVENTS_ERROR

Address offset: 0x104

ECB block encrypt aborted because of a STOP 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_ERROR			ECB block encrypt aborted because of a STOP task or due to an error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.5.3.7 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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.5.3.8 PUBLISH_ERROR

Address offset: 0x184

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.5.3.9 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 W1S	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.5.3.10 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 W1C	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.5.3.11 ERRORSTATUS

Address offset: 0x400

Error 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	ERRORSTATUS			Error status when the ERROR event is generated
			NoError	0	No errors have occurred
			PrematureInptrEnd	1	End of INPTR job list before data structure was read.
			PrematureOutptrEnd	2	End of OUTPTR job list before data structure was read.
			EncryptionTooSlow	3	Encryption aborted due to higher priority peripheral requesting or using the AES module. This enumerator is deprecated.
			Aborted	3	Encryption aborted due to higher priority peripheral requesting or using the AES module.
			DmaError	4	Bus error during DMA access.

8.5.3.12 KEY.VALUE[n] (n=0..3)

Address offset: 0x510 + (n × 0x4)

128-bit 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	VALUE			AES 128-bit key value, bits (32*(n+1))-1 : (32*n)

8.5.3.13 IN

IN EasyDMA channel

8.5.3.13.1 IN.PTR

Address offset: 0x530

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	PTR			Points to a job list containing unencrypted ECB data structure

8.5.3.14 OUT

OUT EasyDMA channel

8.5.3.14.1 OUT.PTR

Address offset: 0x538

Output pointer

Points to a job list containing encrypted ECB 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	PTR			Output pointer

8.5.3.15 CSAA

Channel sounding access address scoring algorithm

8.5.3.15.1 CSAA.REFLECTOR

Address offset: 0x53C

Selected Channel Sounding Access Address used in the CS SYNC from Reflector to Initiator

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

8.5.3.15.2 CSAA.INITIATOR

Address offset: 0x540

Selected Channel Sounding Access Address used in the CS SYNC from Initiator to Reflector

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

8.5.3.15.3 CSAA.MODE

Address offset: 0x544

Operation modes

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

Reverse the endianness on bit level for the ECB output, INITIATOR, and REFLECTOR registers

Enable bit reversal on each byte.

Default endianness

Reversed endianness

8.6 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 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 179 for a list of EGU instances.

8.6.1 Registers

Instances

Instance	Domain	Base address	Description
EGU00	GLOBAL	0x40058000	Event generator unit EGU00
EGU10	GLOBAL	0x40087000	Event generator unit EGU10
EGU20	GLOBAL	0x400C9000	Event generator unit EGU20

Configuration

Instance	Domain	Configuration
EGU00	GLOBAL	6 events
EGU10	GLOBAL	16 events
EGU20	GLOBAL	6 events

Register overview

Register	Offset	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

8.6.1.1 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

8.6.1.2 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
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task TRIGGER[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.6.1.3 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

8.6.1.4 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event TRIGGERED[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.6.1.5 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

8.6.1.6 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 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]
	W1S				
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.6.1.7 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A-P	RW W1C	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

8.7 GPIO — General purpose input/output

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

The number of ports and GPIO pins per port varies with product variant and package. Refer to Registers on page 187 and Pin assignments on page 546 for more information about the number of GPIO pins that are supported.

GPIO has the following user-configurable features:

- Configurable output drive strength
- Internal pull-up and pull-down resistors
- Wake-up from high or low level triggers on all pins in PERI and LP power domains
- Trigger interrupt on state changes on any pin on selected ports, see the wakeup source capability in Port capabilities on page 184
- One or more GPIO outputs can be controlled through PPI and GPIOTE channels
- All pins can be individually mapped to interface blocks for layout flexibility
- GPIO state changes captured on SENSE signal can be stored by LATCH register

The following figure 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.

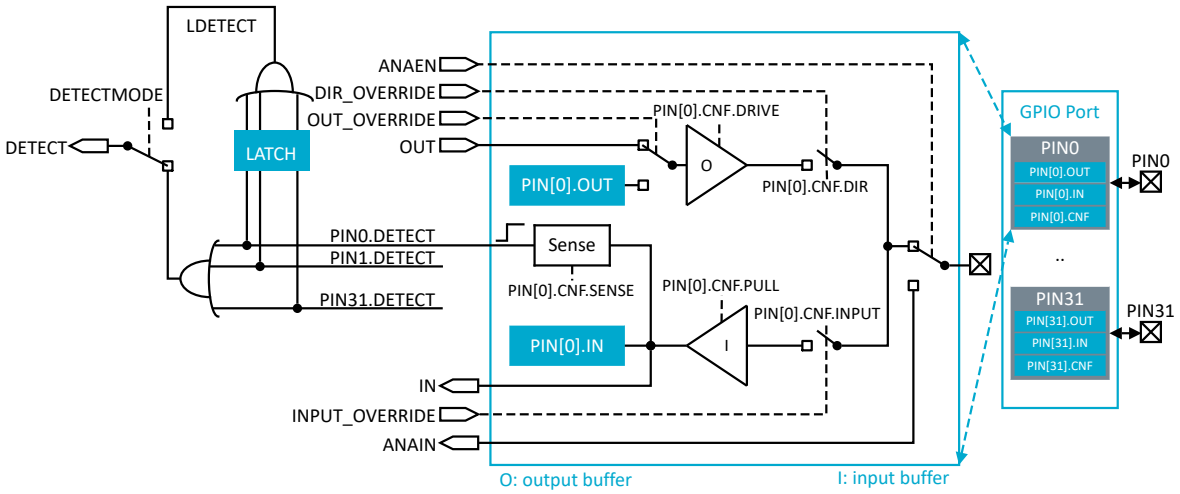


Figure 50: GPIO port and the GPIO pin details

8.7.1 Pin configuration

The GPIO port peripheral implements up to 32 pins, `PIN[n]` (n = 0..31), that can be individually configured in the PIN_CNF[n] registers (n=0..31).

The following parameters can be enabled or configured in these registers:

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

All write-capable registers are retained registers. See POWER — Power control on page 79 for more information.

When not used as an input, disconnect the input buffer of the GPIO pin to save power. An input must be connected to get a valid value in the IN register and for the sense mechanism to have access to the pin.

Other peripherals in the system can connect to GPIO pins to override their output value, override their configuration, or read their analog or digital input value.

Selected pins also support analog input signals (ANAIN). The assignment of the analog pins can be found in Pin assignments on page 546.

GPIO drive strength is configured using the DRIVE0 and DRIVE1 fields of register PIN_CNF[n] (n=0..31) (Retained) on page 191. Some pins may not support every drive configuration, see Pin assignments on page 546 for more information.

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

For more information on pin assignment and the corresponding effect of read and write operations of GPIO registers, see Peripheral and subsystem assignment on page 185.

8.7.2 Pin sense mechanism

Pin sensitivity can be individually configured through the SENSE field in the PIN_CNF[n] register to detect a high level or a low level on their input. When the correct level is detected, the sense mechanism will set the DETECT signal `high`. Each pin has a separate DETECT signal.

The default behavior for the DETECT signal is defined by the register DETECTMODE. By default, the DETECT signals from all pins in the GPIO port are combined into one common DETECT signal that is routed throughout the system, and can be utilized by other peripherals. This mechanism is functional in both System ON and System OFF modes.

Pins must be in a level that cannot trigger the sense mechanism before enabling it. When the sense mechanism is enabled, the DETECT signal will immediately go high if the SENSE condition configured in the PIN_CNF registers is met. This will trigger a PORT event if the DETECT signal was low before enabling the sense mechanism.

The DETECT signal is used by the power and clock management system to exit from System OFF mode, and by the GPIOTE peripheral to allow pins to generate events and interrupts.

When a pin's PINx.DETECT signal goes high, a flag will be set in the register LATCH. For example, when the PIN0.DETECT signal goes high, bit 0 in the register LATCH will be set to `1`. If the CPU performs a clear operation on a bit in the register LATCH when the associated PINx.DETECT signal is high, the bit in the register LATCH will not be cleared. The register LATCH will only be cleared if the CPU explicitly clears it by writing a `1` to the bit to be cleared. This means the register LATCH 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 register LATCH `1`. The LDETECT signal will be set low when all bits in the register LATCH are successfully cleared to `0`.

If one or more bits in the register LATCH are 1 after the CPU has performed a clear operation, a rising edge will be generated on the LDETECT signal. This is illustrated in DETECT signal behavior on page 184.

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

LDETECT is enabled using the DETECTMODE register. See GPIO port and the GPIO pin details on page 182.

The following figure illustrates the DETECT signal behavior for these two alternatives.

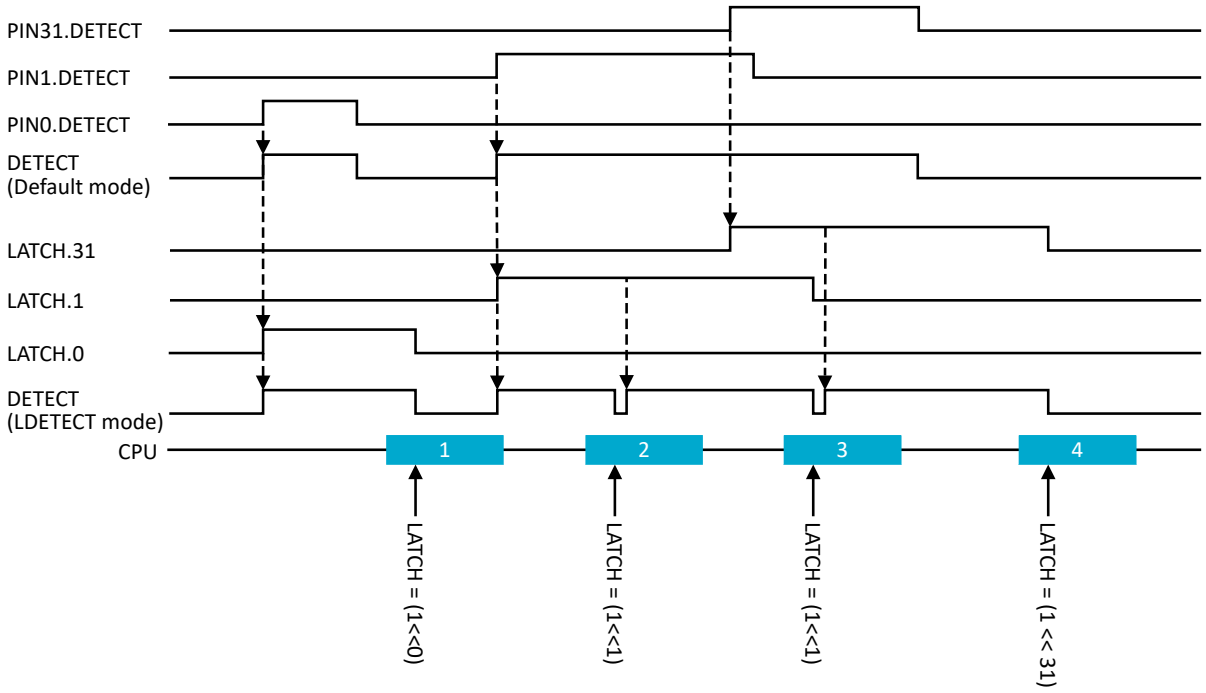


Figure 51: DETECT signal behavior

8.7.3 Port capabilities

The device power domains have their own GPIO ports with different capabilities.

The following is a list of all GPIO ports (P[n]) in the system.

- P0 low-power domain – These I/O pins can wake the system up from System OFF or System ON sleep, and can be accessed by all peripherals in the low-power domain.
- P1 peripheral domain – These I/O pins can wake the system up from System OFF or System ON sleep, and can be accessed by all peripherals in the peripheral domain.

Peripherals cannot mix pins from different ports. All pins must be on the same port.

The following table lists the port special functions and characteristics.

Port	Wakeup source	Extra high drive strength (E0E1)	Pin sense/ detect	GPIOTE	Maximum speed [MHz]
P0	Yes	No	Yes	Yes	8
P1	Yes	No	Yes	Yes	8

Table 30: Port capabilities

Please note that certain peripherals can operate the GPIOs at higher frequencies than those specified in the table above. In addition to the capabilities of the port, some specific pins have additional functions. These are listed in Pin assignments on page 546.

8.7.4 Peripheral and subsystem assignment

System GPIO pins can be allocated to peripherals with dedicated pins or subsystems such as trace and debug.

The pins of the system are listed in Pin assignments on page 546.

A pin can be assigned to any of the following:

- GPIO or peripheral with PSEL registers
- Peripheral with dedicated pins (VPR and GRTC)
- Trace and debug (TND) subsystem

By default, all pins are assigned to GPIO or peripherals with PSEL registers. This is the default value of the CTRLSEL value in register PIN_CNF[n] (n=0..31) (Retained) on page 191.

To allocate a pin to a peripheral or subsystem with dedicated pins, such as GRTC or TND, change the CTRLSEL value in register PIN_CNF[n] (n=0..31) (Retained) on page 191 for that pin. This will connect the pin to the subsystem or peripheral.

Only the peripheral or subsystem where the pin was allocated can observe and control that pin's state. Reading a pin that is not allocated to the current subsystem 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 GPIO peripheral.

When CTRLSEL is used to allocate a peripheral or subsystem, reading the GPIO peripheral registers will not reveal the state of the pins.

The following figure illustrates how to assign a pin to a peripheral that has dedicated pins, or a subsystem such as trace and debug.

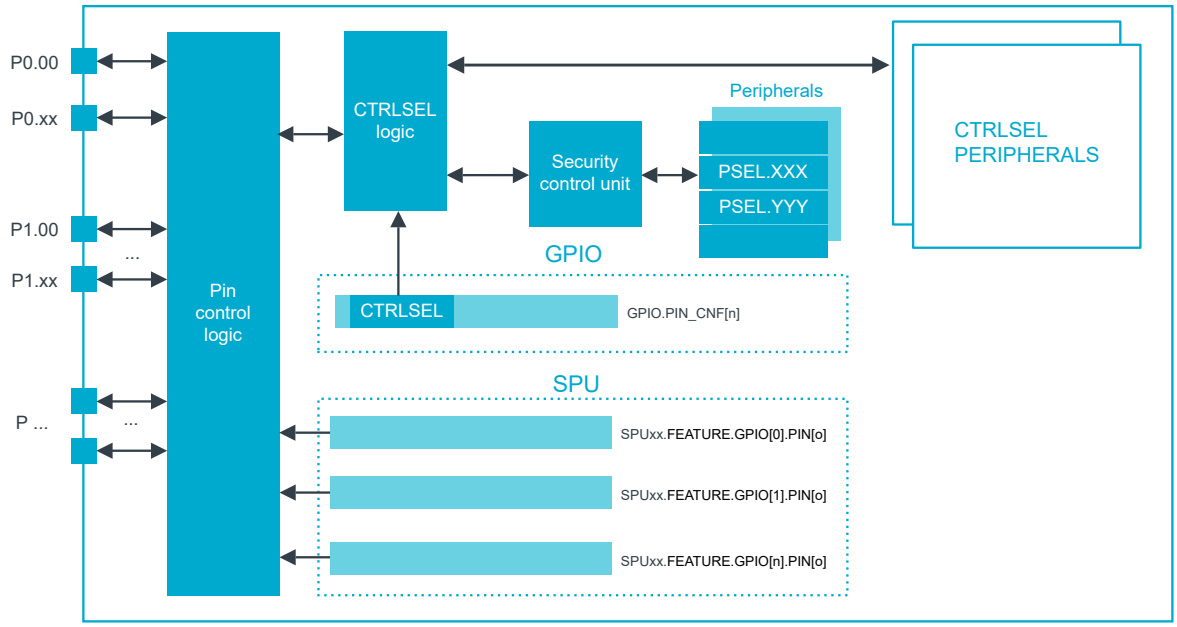


Figure 52: Pin access using CTRLSEL

Note: CTRLSEL must be configured before any pins are used, otherwise glitches on the GPIO pins of the corresponding port can occur.

8.7.5 Clock pins

The device has dedicated clock pins. Some peripherals, such as SPI, TWI, and TRACE, have clock signals.

The dedicated clock pins are optimized to ensure correct timing between the clock and data signals for these peripherals. All peripherals that have clock signals must use these pins. See Pin assignments on page 546 for the full list.

The data signal associated with the peripheral must use pins close to the clock pin. This ensures that the internal paths from the peripheral to the pin have the same delay, so that the data and clock signals reach the pins at the same time.

For high-speed signals, the printed circuit board (PCB) layout must use short PCB traces of identical length. This reduces delays and ensures the same delay on the clock and data path.

8.7.6 Weak pull-ups

A subset of the GPIO pins support weak pull-ups.

When enabled, the weak pull-ups are enabled for the complete set of GPIO pins connected to the weak pull-up module. Unlike GPIO, there is no per-pin configuration. Once enabled, the weak pull-up is applied to all GPIO pins that have weak pull-ups.

The set of pins that are connected to the weak pull-up module can be found in Pin assignments on page 546.

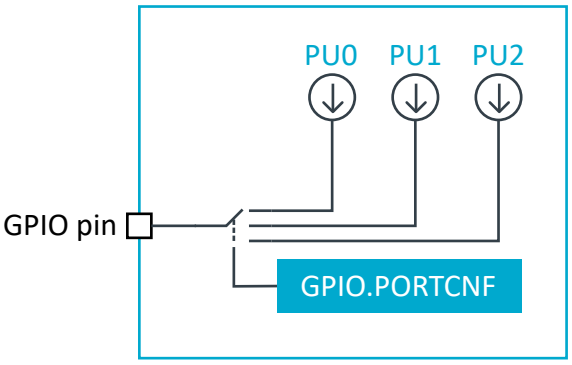


Figure 53: Block diagram

The pull-up is enabled on the full set of pins using PORTCNF.WEAKPU on page 191.

Note: Per-pin configurable pull-up and pull-down resistors are provided by GPIO — General purpose input/output on page 182. They operate independently of the weak pull-up.

8.7.7 Reset behavior

While the **nRESET** pin is asserted (falling edge and held low), GPIO pins will enter one of two states:

- Retain their current configuration (e.g., outputs remain outputs, inputs remain inputs)
- Switch to high impedance, which is the default state upon device startup

Once **nRESET** is deasserted (rising edge), all GPIOs transition to the high impedance state until device firmware configures them.

When GPIO pins are in a high-impedance state for a prolonged period, external circuits that depend on a clearly defined logic level (either high or low) might require the use of external pull-up or pull-down resistors to ensure reliable operation. However, in most cases this is not required, as the duration of the high-impedance state during the reset cycle is typically very short.

8.7.8 Registers

Instances

Instance	Domain	Base address	Description
P1	GLOBAL	0x400D8200	General purpose input and output, port P1
P0	GLOBAL	0x4010A000	General purpose input and output, port P0

Configuration

Instance	Domain	Configuration
P1	GLOBAL	I/O pins on this port have pin sense mechanism This port has weak pull-ups on selected GPIO pins. P1 has 32 pins, P1.00 through P1.31.
P0	GLOBAL	I/O pins on this port have pin sense mechanism P0 has 5 pins, P0.00 through P0.04.

Register overview

Register	Offset	Description
OUT	0x000	Write GPIO port This register is retained.
OUTSET	0x004	Set individual bits in GPIO port
OUTCLR	0x008	Clear individual bits in GPIO port
IN	0x00C	Read GPIO port
DIR	0x010	Direction of GPIO pins This register is retained.
DIRSET	0x014	DIR set register
DIRCLR	0x018	DIR clear register
LATCH	0x020	Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers This register is retained.
DETECTMODE	0x024	Select between default DETECT signal behavior and LDETECT mode This register is retained.
PORTCNF.WEAKPU	0x30	Control global weak pull-ups on the selected set of GPIO pins
PIN_CNF[n]	0x080	Pin n configuration of GPIO pin This register is retained.

8.7.8.1 OUT (Retained)

Address offset: 0x000

Write GPIO port

This register is retained.

Bit number																															
ID																															
Reset 0x00000000																															
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																										

8.7.8.2 OUTSET

Address offset: 0x004

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																															
A-f	RW	PIN[i] (i=0..31)																																	
	W1S																																		
			Low	0																															
			High	1																															
			Set	1																															

Write: writing a '1' sets the pin high; writing a '0' has no effect

8.7.8.3 OUTCLR

Address offset: 0x008

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																															
A-f	RW	PIN[i] (i=0..31)																																	
	W1C																																		
			Low	0																															
			High	1																															
			Clear	1																															

Write: writing a '1' sets the pin low; writing a '0' has no effect

8.7.8.4 IN

Address offset: 0x00C

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																															
A-f	R	PIN[i] (i=0..31)																																	
			Low	0																															
			High	1																															

Pin input is high

8.7.8.5 DIR (Retained)

Address offset: 0x010

Direction of GPIO pins

This register is retained.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				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																															
A-f	RW	PIN[i] (i=0..31)																																	
			Input	0																															
			Output	1																															

8.7.8.6 DIRSET

Address offset: 0x014

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																															
A-f	RW	PIN[i] (i=0..31)																																	
	W1S																																		
			Input	0																															
			Output	1																															
			Set	1																															

8.7.8.7 DIRCLR

Address offset: 0x018

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																															
A-f	RW	PIN[i] (i=0..31)																																	
	W1C																																		
			Input	0																															
			Output	1																															
			Clear	1																															

8.7.8.8 LATCH (Retained)

Address offset: 0x020

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

This register is retained.

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

8.7.8.9 DETECTMODE (Retained)

Address offset: 0x024

Select between default DETECT signal behavior and LDETECT mode

This register is retained.

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

8.7.8.10 PORTCNF.WEAKPU

Address offset: 0x30

Control global weak pull-ups on the selected set 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				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	PULLUP			Control pull-up.
			Disabled	0	Pull-up is disabled.
			PU0	1	Enable pull-up 0.
			PU1	2	Enable pull-up 1.
			PU2	3	Enable pull-up 2.

8.7.8.11 PIN_CNF[n] (n=0..31) (Retained)

Address offset: 0x080 + (n × 0x4)

Pin n configuration of GPIO pin

This register is retained.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G G G F F E E 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
A	RW	DIR			Pin direction. Same physical register as DIR register
			Input	0	Configure pin as an input pin
			Output	1	Configure pin as an output pin

Bit number	31	30	29	28	27	26	25	24	23	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											F	F					E	E	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
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	DRIVE0			Drive configuration for '0'
			S0	0	Standard '0'
			H0	1	High drive '0'
			D0	2	Disconnect '0'(normally used for wired-or connections)
			E0	3	Extra high drive '0'
					Note: The DRIVE1 must be E1 as well to work properly.
E	RW	DRIVE1			Drive configuration for '1'
			S1	0	Standard '1'
			H1	1	High drive '1'
			D1	2	Disconnect '1'(normally used for wired-or connections)
			E1	3	Extra high drive '1'
					Note: The DRIVE0 must be E0 as well to work properly.
F	RW	SENSE			Pin sensing mechanism
			Disabled	0	Disabled
			High	2	Sense for high level
			Low	3	Sense for low level
G	RW	CTRLSEL			Select which module has direct control over this pin
			GPIO	0x0	GPIO or peripherals with PSEL registers
			VPR	0x1	VPR processor
			GRTC	0x4	GRTC peripheral

8.8 GPIOTE — GPIO tasks and events

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

The main features of GPIOTE are the following:

- GPIO pin state change by triggering tasks
- Event generation on GPIO pin state change
- PORT event generation on GPIO DETECT signal

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. See Peripheral interface on page 138 for additional information on tasks and events. GPIO is described in more detail in GPIO — General purpose input/output on page 182.

Detection of pin state changes when in low power mode 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. Tasks SET and CLR are fixed. OUT can be configured 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

8.8.1 Pin events and tasks

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

The tasks SET[n], CLR[n], and OUT[n] can 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 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 therefore 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 31: 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].

8.8.2 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 182 for more information about the DETECT signal.

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. Make sure to 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 183 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.

In order to prevent spurious interrupts from the PORT event while configuring the sources, the following must be performed:

1. Disable interrupts on the PORT event (through INTENCLR.PORT).
2. Configure the sources (PIN_CNF[n].SENSE in GPIO).
3. Clear any potential event that could have occurred during configuration (write '0' to EVENTS_PORT).
4. Enable interrupts (through INTENSET.PORT).

8.8.3 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.

For the range of possible CONFIG.PORT values in the product, see Instances on page 194. Writing other values may lead to undefined behavior.

Note: A pin can only be assigned to one GPIOTE channel at a time. Failing to do so may result in unpredictable behavior.

8.8.4 Registers

Instances

Instance	Domain	Base address	Description
GPIOTE20	GLOBAL	0x400DA000	8 channels and 1 interrupts for GPIO port P1 GPIO tasks and events GPIOTE20
GPIOTE30	GLOBAL	0x4010C000	4 channels and 1 interrupts for GPIO port P0 GPIO tasks and events GPIOTE30

Configuration

Instance	Domain	Configuration
GPIOTE20	GLOBAL	Number of GPIOTE channels: 0..7 Number of GPIOTE port events: 0..0 Number of GPIOTE interrupts: 0..0
GPIOTE30	GLOBAL	Number of GPIOTE channels: 0..3 Number of GPIOTE port events: 0..0 Number of GPIOTE interrupts: 0..0

Register overview

Register	Offset	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 from pin specified in CONFIG[n].PSEL
EVENTS_PORT[n].NONSECURE	0x140	Port event (the event name NONSECURE was chosen to be API compatible with devices that feature Arm TrustZone security)
PUBLISH_IN[n]	0x180	Publish configuration for event IN[n]
PUBLISH_PORT[n].NONSECURE	0x1C0	Publish configuration for event PORT[n].NONSECURE
INTENSET0	0x304	Enable interrupt
INTENCLR0	0x308	Disable interrupt
INTENSET1	0x314	Enable interrupt
INTENCLR1	0x318	Disable interrupt
CONFIG[n]	0x510	Configuration for OUT[n], SET[n], and CLR[n] tasks and IN[n] event

8.8.4.1 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

8.8.4.2 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

8.8.4.3 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**

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

8.8.4.4 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task OUT[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.8.4.5 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task SET[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.8.4.6 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CLR[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.8.4.7 EVENTS_IN[n] (n=0..7)

Address offset: 0x100 + (n × 0x4)

Event 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 from pin specified in CONFIG[n].PSEL
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.8.4.8 EVENTS_PORT[n] (n=0..0)

Peripheral events.

8.8.4.8.1 EVENTS_PORT[n].NONSECURE (n=0..0)

Address offset: 0x140 + (n × 0x8)

Port event (the event name NONSECURE was chosen to be API compatible with devices that feature Arm TrustZone security)

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

ID	R/W	Field	Value ID	Value	Description
A	RW	NONSECURE			Port event (the event name NONSECURE was chosen to be API compatible with devices that feature Arm TrustZone security)
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.8.4.9 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event IN[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.8.4.10 PUBLISH_PORT[n] (n=0..0)

Publish configuration for events

8.8.4.10.1 PUBLISH_PORT[n].NONSECURE (n=0..0)

Address offset: 0x1C0 + (n × 0x8)

Publish configuration for event PORT[n].NONSECURE

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event PORT[n].NONSECURE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.8.4.11 INTENSET0

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 W1S	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 W1S	PORT0NONSECURE			Write '1' to enable interrupt for event PORT0NONSECURE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.8.4.12 INTENCLR0

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 W1C	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 W1C	PORT0NONSECURE			Write '1' to disable interrupt for event PORT0NONSECURE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.8.4.13 INTENSET1

Address offset: 0x314

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																															
A-H	RW W1S	IN[i] (i=0..7)																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
I	RW W1S	PORT0NONSECURE																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															

8.8.4.14 INTENCLR1

Address offset: 0x318

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																															
A-H	RW W1C	IN[i] (i=0..7)																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
I	RW W1C	PORT0NONSECURE																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															

8.8.4.15 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	C	C	C	B	B	B	B	B			A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value																															
A	RW	MODE																																	
			Disabled	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															E			D	D				C	C	C	B	B	B	B	B			A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
			Event	1	Event mode The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.
			Task	3	Task mode The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.
B	RW	PSEL		[0..31]	GPIO number associated with SET[n], CLR[n], and OUT[n] tasks and IN[n] event
C	RW	PORT		[0..15]	Port number
D	RW	POLARITY			When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.
			None	0	Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.
			LoToHi	1	Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.
			HiToLo	2	Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.
			Toggle	3	Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.
E	RW	OUTINIT			When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.
			Low	0	Task mode: Initial value of pin before task triggering is low
			High	1	Task mode: Initial value of pin before task triggering is high

8.9 GRTC — Global real-time counter

The global real-time counter peripheral (GRTC) is an ultra-low power shared system timer. GRTC implements a high-resolution system timer that is available in all power modes, including System OFF.

The system timer has a 1 µs resolution and is 52 bits wide. This provides a run time of 142 years after initial power-on until the counter wraps around. It uses the 16 MHz clock when the high-speed clock is active, but automatically switches to 32.768 kHz in the other power modes. It will continue to be updated in all power modes. Due to the combination of clock sources, it has a 1 µs resolution and an accuracy equal to the 32.768 kHz clock.

The main features of GRTC are the following:

- System timer – SYSCOUNTER
  - 1 µs resolution
  - Multiple compare/capture channels on SYSCOUNTER
  - PPI connection
  - Periodic interval generation for one compare event
  - Wake up from System OFF mode
- Pulse Width Modulation (PWM)

- Single channel PWM
- Operates in System OFF mode

- Clock output on pin
  - LFCLK
  - Configurable divided fast clock output

The system timer is a high resolution timer which can be accessed by all processors in the system. It allows the same system counter to be shared among all users. GRTC can operate in System OFF mode.

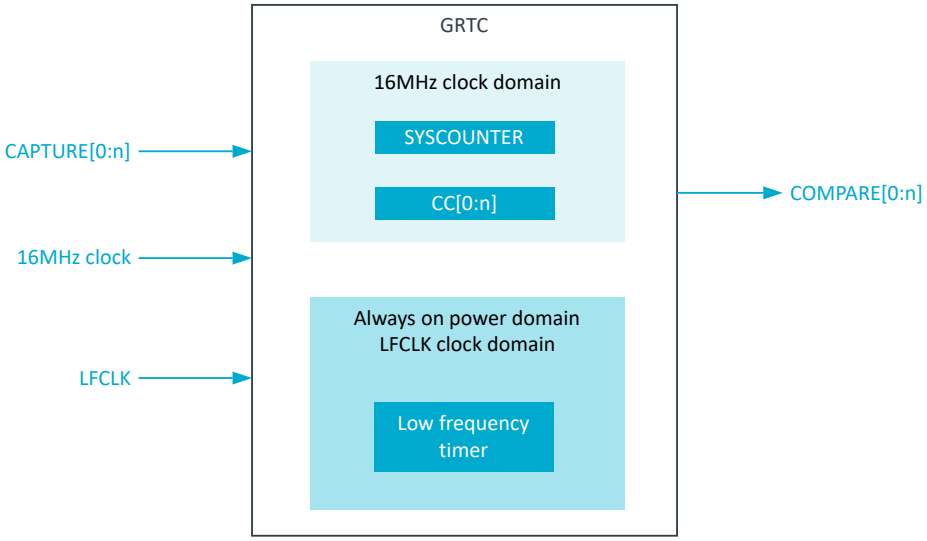


Figure 54: GRTC block diagram

The internal low-frequency timer can run while in System OFF mode.

All GRTC registers are reset during wakeup from System OFF mode. However, the SYSCOUNTER[m].SYSCOUNTERL and SYSCOUNTER[m].SYSCOUNTERH registers are restored automatically on wakeup from System OFF mode and after soft reset.

8.9.1 GRTC clock sources

The low frequency timer in GRTC will run off the LFCLK.

When the low frequency timer is started, the GRTC peripheral will automatically request the LFCLK source if the LFCLK is not already running.

The GRTC low frequency timer clock source can be selected using register CLKCFG.CLKSEL. The clock source cannot be changed after GRTC is started.

The clock source selection at GRTC is retained internally during System OFF mode, even though the CLKCFG.CLKSEL is reset on wakeup.

The SYSCOUNTER runs off the fast clock (16MHz clock), but uses only LFCLK while SYSCOUNTER is in sleep state.

See CLOCK for more information about clock sources.

8.9.2 SYSCOUNTER

The internal counter at SYSCOUNTER increments every 1 µs.

SYSCOUNTER is a 52-bit counter and is enabled using MODE.SYSCOUNTEREN register. The internal low frequency timer must be started for the proper operation of the SYSCOUNTER while SYSCOUNTER goes into sleep mode. It can be started by the TASKS_START task and can be stopped by the TASKS_STOP task when SYSCOUNTER is no more in use.

The current value of SYSCOUNTER can be read using corresponding SYSCOUNTER[m].SYSCOUNTERL and SYSCOUNTER[m].SYSCOUNTERH. But, the SYSCOUNTER[m].SYSCOUNTERL must be read before corresponding SYSCOUNTER[m].SYSCOUNTERH. The SYSCOUNTER[m].SYSCOUNTERH.OVERFLOW indicates if the SYSCOUNTER[m].SYSCOUNTERL is overflown after reading it.

Sample code for reading the SYSCOUNTER value:

```
    uint32_t syscounterl_value, syscounterh_value, syscounterh;
    uint64_t syscounter;

    do
    {
      syscounterl_value = GRTC.SYSCOUNTER[m].SYSCOUNTERL;
      syscounterh = GRTC.SYSCOUNTER[m].SYSCOUNTERH;

      syscounterh_value = ((syscounterh & GRTC_SYSCOUNTER_SYSCOUNTERH_VALUE_Msk) >>
GRTC_SYSCOUNTER_SYSCOUNTERH_VALUE_Pos);

      if (((syscounterh & GRTC_SYSCOUNTER_SYSCOUNTERH_OVERFLOW_Msk) >>
GRTC_SYSCOUNTER_SYSCOUNTERH_OVERFLOW_Pos) ==
GRTC_SYSCOUNTER_SYSCOUNTERH_OVERFLOW_Overflow)
      {
        syscounterh = syscounterh_value - 1;
      }

    } while (((syscounterh & GRTC_SYSCOUNTER_SYSCOUNTERH_BUSY_Msk) >>
GRTC_SYSCOUNTER_SYSCOUNTERH_BUSY_Pos) != GRTC_SYSCOUNTER_SYSCOUNTERH_BUSY_Ready);

    syscounter = (syscounterh_value << 32) + syscounterl_value;
```

Compare and Capture (CC)

The CC[n] is a group of registers interfacing the compare and capture channels of SYSCOUNTER, where *n* is the number of compare and capture channels specified in the GRTC instance configuration table below. Each CC[n] has an associated TASKS_CAPTURE[n] task and EVENTS_COMPARE[n] event. Each channel compare functionality can be enabled/disabled with CC[n].CCEN.ACTIVE, and its current active state read from this same register.

The EVENTS_COMPARE[n] event can be generated by writing the compare values to the corresponding CC[n].CCL and CC[n].CCH registers. When a channel compare functionality is enabled and SYSCOUNTER is equal or greater than that CC value EVENTS_COMPARE[n] will be generated. Writes to CC[n].CCL disable the corresponding compare channel and writes to CC[n].CCH enable it. So CC[n].CCL must be written first.

Each channel compare functionality, except for the first channel when operating in periodic mode, operates in one-shot mode. When operating in one-shot mode, CC[n].CCEN.ACTIVE is cleared automatically following a compare event.

Moreover, a compare channel is automatically disabled when triggering TASKS_CAPTURE[n].

The EVENTS_COMPARE[n] event is generated immediately if the configured compare value at CC[n] is less than the current SYSCOUNTER value. The status CC[n].CCEN.PASTCC indicates this condition.

Every time the TASKS_CAPTURE[n] task is triggered, the current SYSCOUNTER is copied into the corresponding CC[n].CCL and CC[n].CCH registers. The CC[n].CCL and CC[n].CCH registers can be read in any order. The TASKS_CAPTURE[n] task will not generate EVENTS_COMPARE[n] event.

The TASKS_CAPTURE[n] tasks and EVENTS_COMPARE[n] events can be connected with the PPI. However, the TASKS_CAPTURE[n] is functional only when the SYSCOUNTER is in active state. The GRTC can be forced into active state by setting any SYSCOUNTER[n].ACTIVE register.

The compare value for CC[n] can also be updated by adding a fixed value provided at CC[n].CCADD.VALUE. Based on the CC[n].CCADD.REFERENCE configuration, either the current CC value or the current SYSCOUNTER value is added to the CC[n].CCADD.VALUE to configure the new compare value. If the CC[n] overflows after writing to CC[n].CCADD.VALUE, then EVENTS_COMPARE[n] is generated immediately. Writing to CC[n].CCADD enables the corresponding compare channel.

Writes to CC[n].CCADD are ignored when the SYSCOUNTER is in sleep state.

Periodic interval

In addition to one-shot mode, the CC[0] can produce periodic EVENTS_COMPARE[0] event without any software interaction. The interval between these events can be programmed using INTERVAL register and non-zero interval enables this periodic interval feature. On every EVENTS_COMPARE[0] event, CC[0] becomes CC[0] + INTERVAL.

SYSCOUNTER sleep mode

SYSCOUNTER supports the following power modes:

- SYSCOUNTER is in active state
- SYSCOUNTER is in sleep state - This is the GRTC ultra-low power sleep mode

To save power, SYSCOUNTER automatically goes into sleep state when there is no activity.

Before SYSCOUNTER goes into sleep state, the GRTC configures the internal low frequency timer compare match based on the next expected SYSCOUNTER compare match using CC[n] configuration. An internal event on low frequency timer is generated when the compare match happens.

The internal counter at SYSCOUNTER is not ticking when SYSCOUNTER is in sleep state, the internal low frequency timer is configured as described above.

SYSCOUNTER returns to active state when any one of the following condition is met,

- Any of SYSCOUNTER[n].ACTIVE register is set to Active
- SYSCOUNTER counter value is read
- Any of the following registers are written:
  - MODE
  - CC[n]
  - INTENn/INTENSETn/INTENCLRn/INTENPENDn
  - Write to tasks TASKS_START, TASKS_STOP and TASKS_CLEAR
  - When internal low frequency timer compare match happens
- Any CPU is not sleeping, if MODE.AUTOEN is set

SYSCOUNTER goes back into sleep state when none of the above conditions met. However, the SYSCOUNTER active state can be extended by configuring the number of LFCLK cycles at TIMEOUT register.

On wake up to active state, SYSCOUNTER is updated based on the internal low frequency timer compare match. The status SYSCOUNTER[m].SYSCOUNTERH.BUSY indicates SYSCOUNTER is synchronized and valid after the SYSCOUNTER is woken up.

The status SYSCOUNTER[m].SYSCOUNTERH.LOADED indicates SYSCOUNTER is loaded after wake up, but not synchomized to the low frequency clock yet. This gives early indication of the SYSCOUNTER value, whcih may vary approximately one low frequency clcok cycle.

The SYSCOUNTER status can be summerized as below:

- SYSCOUNTER[m].SYSCOUNTERH.BUSY = 1 and SYSCOUNTER[m].SYSCOUNTERH.LOADED = 0 : SYSCCOUNTER is in sleep and SYSCOUNTER[m].SYSCOUNTER value is invalid
- SYSCOUNTER[m].SYSCOUNTERH.BUSY = 1 and SYSCOUNTER[m].SYSCOUNTERH.LOADED = 1 : SYSCCOUNTER is in woken up and SYSCOUNTER[m].SYSCOUNTER value is updated but not synchronized to low frequency clock yet
- SYSCOUNTER[m].SYSCOUNTERH.BUSY = 0 and SYSCOUNTER[m].SYSCOUNTERH.LOADED = 1 : SYSCOUNTER[m].SYSCOUNTER value is valid

Before handling next scheduled EVENTS_COMPARE[n] event, GRTC must wake up from the low power state. The WAKETIME register configures the number of LFCLK cycles that GRTC will wake up before the compare event. This duration allows the device sufficient time to power up, initialize, and activate the necessary clocks to accurately generate and handle events from the GRTC SYSCOUNTER. A longer system wake-up time requires a larger WAKETIME value to ensure reliable event processing. When the device is in System OFF, the WAKETIME must cover the system wakeup time from System OFF mode and in addition the required time the system uses to configure GRTC and enable the event.

All GRTC registers must be restored at wakeup from system OFF before the next scheduled COMPARE[n] event is generated. The TIMEOUT register must be configured to a value higher than WAKETIME (TIMEOUT > WAKETIME + guard_time). This makes sure that GRTC is not entering sleep again if the next event is nearer than TIMEOUT LFCLK cycles. The minimum guard time is 1 LFCLK cycle.

Recommendation on reading SYSCOUNTER

The following steps are recommended while reading SYSCOUNTER:

1. Set the corresponding SYSCOUNTER[m].ACTIVE to Active
2. Wait until the corresponding status SYSCOUNTER[m].SYSCOUNTERH.BUSY is cleared
3. Read the corresponding SYSCOUNTER[m].SYSCOUNTERL/H values
4. Clear the SYSCOUNTER[m].ACTIVE set above

Entering System OFF mode

The following steps are recommended before entering System OFF mode:

1. Set the SYSCOUNTER in active state, either
   - Set MODE.AUTOEN
   - Set corresponding SYSCOUNTER[m].ACTIVE to Active
2. If GRTC is wakeup source, then set the corresponding CC[n] value to expected wakeup time
3. Set WAKETIME for the boot latency
4. Set the SYSCOUNTER in sleep state, by clearing the configuration set at step 1 above
5. Wait for either of EVENTS_RTCOMPARESYNC or any EVENTS_COMPARE[n]
   - If any EVENTS_COMPARE[n] triggered,
     a. Allow CPUs to wakeup on interrupts
     b. Do not enter System OFF mode
   - Else,
     a. Enter System OFF by using the SYSTEMOFF register

8.9.3 Pulse Width Modulation (PWM)

The GRTC peripheral has a built-in PWM that can drive one output pin as an 8-bit non-inverted pulse-width modulated output.

The PWM is based on the internal low frequency timer of the GRTC and the PWM has a period of 256 LFCLK clock cycles, resulting frequency is 128Hz.

The PWM can be started by the TASKS_PWMSTART task and can be stopped by the TASKS_PWMSTOP task. The PWM starts/stops on next time when the lower 8 bits of internal low frequency timer becomes to zero. It takes up to 256 LFCLK clock cycles to take these tasks to go into effect.

The PWM compare value is configured using PWMCONFIG and the copied to the internal PWM compare register when the lower 8 bits of internal low frequency timer is 0.

The PWM output goes high when the the lower 8 bits of internal low frequency timer goes to zero and the PWM output goes low when the lower 8 bits of internal low frequency timer matches the PWM compare value. The EVENTS_PWMPERIODEND event is generated on the rising edge of the PWM output.

To optimize the GRTC power consumption, the EVENTS_PWMPERIODEND can be disabled using EVTEN/EVTENSET/EVTENCLR registers to prevent clock from being requested when those events are triggered.

The PWM is operating even while the device is in system OFF.

For the PWM output pin mapping, see Pin assignments on page 546. GRTC uses standard output drive strength for PWM output.

8.9.4 Clock output

The GRTC can be configured to output the clock on a pin.

The following clocks can be configured to be output on pins:

- LFCLK (32 KHz clock) output
- Divided 16M Hz clock (fast clock) output

The clock outputs can be enabled or disabled using CLKOUT.

The 16 MHz clock is divided before it is output on a pin. The divisor can be configured in CLKCFG, where clock output is (fast clock) / (CLKCFG.CLKFASTDIV * 2). The CLKCFG.CLKFASTDIV should be changed only when CLKOUT.CLKOUTFAST is disabled.

The LFCLK clock is output also when the device is in System OFF mode.

For the clock output pin mapping, see Pin assignments on page 546. GRTC uses standard output drive strength for clock output.

8.9.5 Task priority

If the START task and the STOP task are triggered at the same time, meaning within the same period of LFCLK, 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 LFCLP, the CAPTURE tasks are prioritized. This means that the CC registers will capture the counter value before the CLEAR tasks are triggered.

8.9.6 Registers

Instances

Instance	Domain	Base address	Description
GRTC	GLOBAL	0x400E2000	Global RTC GRTC

Configuration

Instance	Domain	Configuration
GRTC	GLOBAL	Local CPUs connected to MODE.AUTO.EN.CpuActive : Application core Arm Cortex-M33. CLKSEL settings XO and LFLPRC must be used for lowest possible power consumption in sleep modes. Width of the RTCOUNTERH, RTCOMPAREH and RTCOMPARESYNCH registers : 0..14 Number of compare/capture registers : 0..11 Width of the TIMEOUT register : 0..15 Number of GRTC interrupts : 0..3 The PWM registers are available. The CLKOUT register is available. The CLKCFG.CLKSEL register is available. The CLKCFG.CLKSEL register supports LFLPRC. The CC[n].CCADD register has write access only. The ready status and events are available. SYSCOUNTER[n].SYSCOUNTERH.LOADED status is available CC[n].CCEN.PASTCC status is available 4 interrupts with interrupt remapping 12 capture compare channels implemented

Register overview

Register	Offset	Description
TASKS_CAPTURE[n]	0x000	Capture the counter value to CC[n] register
TASKS_START	0x060	Start the counter
TASKS_STOP	0x064	Stop the counter
TASKS_CLEAR	0x068	Clear the counter
TASKS_PWMSTART	0x06C	Start the PWM
TASKS_PWMSTOP	0x070	Stop the PWM
SUBSCRIBE_CAPTURE[n]	0x080	Subscribe configuration for task CAPTURE[n]
EVENTS_COMPARE[n]	0x100	Compare event on CC[n] match
EVENTS_RTCOMPARESYNC	0x164	The GRTC low frequency timer is synchronized with the SYSCOUNTER
EVENTS_PWMPERIODEND	0x16C	Event on end of each PWM period
EVENTS_PWMREADY	0x174	Event on STATUS.PWM.READY status changed to ready
EVENTS_CLKOUTREADY	0x178	Event on STATUS.CLKOUT.READY status changed to ready
PUBLISH_COMPARE[n]	0x180	Publish configuration for event COMPARE[n]

Register	Offset	Description
PUBLISH_PWMREADY	0x1F4	Publish configuration for event PWMREADY
PUBLISH_CLKOUTREADY	0x1F8	Publish configuration for event CLKOUTREADY
SHORTS	0x200	Shortcuts between local events and tasks
INTEN0	0x300	Enable or disable interrupt
INTENSET0	0x304	Enable interrupt
INTENCLR0	0x308	Disable interrupt
INTPEND0	0x30C	Pending interrupts
INTEN1	0x310	Enable or disable interrupt
INTENSET1	0x314	Enable interrupt
INTENCLR1	0x318	Disable interrupt
INTPEND1	0x31C	Pending interrupts
INTEN2	0x320	Enable or disable interrupt
INTENSET2	0x324	Enable interrupt
INTENCLR2	0x328	Disable interrupt
INTPEND2	0x32C	Pending interrupts
INTEN3	0x330	Enable or disable interrupt
INTENSET3	0x334	Enable interrupt
INTENCLR3	0x338	Disable interrupt
INTPEND3	0x33C	Pending interrupts
EVTEN	0x400	Enable or disable event routing
EVTENSET	0x404	Enable event routing
EVTENCLR	0x408	Disable event routing
MODE	0x510	Counter mode selection
CC[n].CCL	0x520	The lower 32-bits of Capture/Compare register CC[n]
CC[n].CCH	0x524	The higher 32-bits of Capture/Compare register CC[n]
CC[n].CCADD	0x528	Count to add to CC[n] when this register is written.
CC[n].CCEN	0x52C	Configure Capture/Compare register CC[n]
TIMEOUT	0x6A4	Timeout after all CPUs gone into sleep state to stop the SYSCOUNTER
INTERVAL	0x6A8	Count to add to CC[0] when the event EVENTS_COMPARE[0] triggers.
WAKETIME	0x6AC	GRTC wake up time.
STATUS.LFTIMER	0x6B0	Low frequency timer status.
STATUS.PWM	0x6B4	PWM status.
STATUS.CLKOUT	0x6B8	CLKOUT configuration status.
PWMCONFIG	0x710	PWM configuration.
CLKOUT	0x714	Configuration of clock output
CLKCFG	0x718	Clock Configuration
SYSCOUNTER[n].SYSCOUNTERL	0x720	The lower 32-bits of the SYSCOUNTER for index [n]
SYSCOUNTER[n].SYSCOUNTERH	0x724	The higher 20-bits of the SYSCOUNTER for index [n]
SYSCOUNTER[n].ACTIVE	0x728	Request to keep the SYSCOUNTER in the active state and prevent going to sleep for index [n]

8.9.6.1 TASKS_CAPTURE[n] (n=0..11)

Address offset: 0x000 + (n × 0x4)

Capture the counter 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 the counter value to CC[n] register
			Trigger	1	Trigger task

8.9.6.2 TASKS_START

Address offset: 0x060

Start the 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 the counter
			Trigger	1	Trigger task

8.9.6.3 TASKS_STOP

Address offset: 0x064

Stop the 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 the counter
			Trigger	1	Trigger task

8.9.6.4 TASKS_CLEAR

Address offset: 0x068

Clear the 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 the counter
			Trigger	1	Trigger task

8.9.6.5 TASKS_PWMSTART

Address offset: 0x06C

Start the PWM

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

8.9.6.6 TASKS_PWMSTOP

Address offset: 0x070

Stop the PWM

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

8.9.6.7 SUBSCRIBE_CAPTURE[n] (n=0..11)

Address offset: 0x080 + (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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CAPTURE[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.9.6.8 EVENTS_COMPARE[n] (n=0..11)

Address offset: 0x100 + (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

8.9.6.9 EVENTS_RTCOMPARESYNC

Address offset: 0x164

The GRTC low frequency timer is synchronized with the SYSCOUNTER

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_RTCOMPARESYNC			The GRTC low frequency timer is synchronized with the SYSCOUNTER
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.9.6.10 EVENTS_PWMPERIODEND

Address offset: 0x16C

Event on 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			Event on end of each PWM period
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.9.6.11 EVENTS_PWMREADY

Address offset: 0x174

Event on STATUS.PWM.READY status changed 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	EVENTS_PWMREADY			Event on STATUS.PWM.READY status changed to ready
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.9.6.12 EVENTS_CLKOUTREADY

Address offset: 0x178

Event on STATUS.CLKOUT.READY status changed 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	EVENTS_CLKOUTREADY			Event on STATUS.CLKOUT.READY status changed to ready
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.9.6.13 PUBLISH_COMPARE[n] (n=0..11)

Address offset: 0x180 + (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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event COMPARE[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.9.6.14 PUBLISH_PWMREADY

Address offset: 0x1F4

Publish configuration for event PWMREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event PWMREADY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.9.6.15 PUBLISH_CLKOUTREADY

Address offset: 0x1F8

Publish configuration for event CLKOUTREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event CLKOUTREADY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.9.6.16 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	RTCOMPARE_CLEAR			Shortcut between event RTCOMPARE and task CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.9.6.17 INTEN0

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-L	RW	COMPARE[i] (i=0..11)			Enable or disable interrupt for event COMPARE[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					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																															
			Disabled	0																															
			Enabled	1																															
M	RW	RTCOMPARESYNC																																	
			Disabled	0																															
			Enabled	1																															
N	RW	PWMPERIODEND																																	
			Disabled	0																															
			Enabled	1																															
O	RW	PWMREADY																																	
			Disabled	0																															
			Enabled	1																															
P	RW	CLKOUTREADY																																	
			Disabled	0																															
			Enabled	1																															

8.9.6.18 INTENSET0

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																															
A-L	RW W1S	COMPARE[i] (i=0..11)																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
M	RW W1S	RTCOMPARESYNC																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
N	RW W1S	PWMPERIODEND																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
O	RW W1S	PWMREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
P	RW W1S	CLKOUTREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															

8.9.6.19 INTENCLR0

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-L	RW W1C	COMPARE[i] (i=0..11)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	RTCOMPARESYNC			Write '1' to disable interrupt for event RTCOMPARESYNC
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	PWMPERIODEND			Write '1' to disable interrupt for event PWMPERIODEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1C	PWMREADY			Write '1' to disable interrupt for event PWMREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1C	CLKOUTREADY			Write '1' to disable interrupt for event CLKOUTREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.9.6.20 INTPEND0

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-L	R	COMPARE[i] (i=0..11)			Read pending status of interrupt for event COMPARE[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
M	R	RTCOMPARESYNC			Read pending status of interrupt for event RTCOMPARESYNC
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
N	R	PWMPERIODEND			Read pending status of interrupt for event PWMPERIODEND
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

Bit number				31 30 29 28 27 26 25 24 23 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
O	R	PWMREADY			Read pending status of interrupt for event PWMREADY
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
P	R	CLKOUTREADY			Read pending status of interrupt for event CLKOUTREADY
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

8.9.6.21 INTEN1

Address offset: 0x310

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-L	RW	COMPARE[i] (i=0..11)			Enable or disable interrupt for event COMPARE[i]
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	RTCOMPARESYNC			Enable or disable interrupt for event RTCOMPARESYNC
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	PWMPERIODEND			Enable or disable interrupt for event PWMPERIODEND
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	PWMREADY			Enable or disable interrupt for event PWMREADY
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	CLKOUTREADY			Enable or disable interrupt for event CLKOUTREADY
			Disabled	0	Disable
			Enabled	1	Enable

8.9.6.22 INTENSET1

Address offset: 0x314

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-L	RW W1S	COMPARE[i] (i=0..11)			Write '1' to enable interrupt for event COMPARE[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	RTCOMPARESYNC			Write '1' to enable interrupt for event RTCOMPARESYNC
			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					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																															
			Disabled	0																															
			Enabled	1																															
N	RW W1S	PWMPERIODEND																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
O	RW W1S	PWMREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
P	RW W1S	CLKOUTREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															

8.9.6.23 INTENCLR1

Address offset: 0x318

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																															
A-L	RW W1C	COMPARE[i] (i=0..11)																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
M	RW W1C	RTCOMPARESYNC																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
N	RW W1C	PWMPERIODEND																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
O	RW W1C	PWMREADY																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
P	RW W1C	CLKOUTREADY																																	

Bit number				31	30	29	28	27	26	25	24	23	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																															
			Clear	1																															
			Disabled	0																															
			Enabled	1																															

8.9.6.24 INTPEND1

Address offset: 0x31C

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																															
A-L	R	COMPARE[i] (i=0..11)																																	
			NotPending	0																															
			Pending	1																															
M	R	RTCOMPARESYNC																																	
			NotPending	0																															
			Pending	1																															
N	R	PWMPERIODEND																																	
			NotPending	0																															
			Pending	1																															
O	R	PWMREADY																																	
			NotPending	0																															
			Pending	1																															
P	R	CLKOUTREADY																																	
			NotPending	0																															
			Pending	1																															

8.9.6.25 INTEN2

Address offset: 0x320

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																															
A-L	RW	COMPARE[i] (i=0..11)																																	
			Disabled	0																															
			Enabled	1																															
M	RW	RTCOMPARESYNC																																	
			Disabled	0																															
			Enabled	1																															
N	RW	PWMPERIODEND																																	
			Disabled	0																															
			Enabled	1																															
O	RW	PWMREADY																																	
			Disabled	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					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																															
			Enabled	1																															
P	RW	CLKOUTREADY																																	
			Disabled	0																															
			Enabled	1																															

8.9.6.26 INTENSET2

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					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																															
A-L	RW W1S	COMPARE[i] (i=0..11)																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
M	RW W1S	RTCOMPARESYNC																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
N	RW W1S	PWMPERIODEND																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
O	RW W1S	PWMREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															
P	RW W1S	CLKOUTREADY																																	
			Set	1																															
			Disabled	0																															
			Enabled	1																															

8.9.6.27 INTENCLR2

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				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-L	RW W1C	COMPARE[i] (i=0..11)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	RTCOMPARESYNC			Write '1' to disable interrupt for event RTCOMPARESYNC
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	PWMPERIODEND			Write '1' to disable interrupt for event PWMPERIODEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1C	PWMREADY			Write '1' to disable interrupt for event PWMREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1C	CLKOUTREADY			Write '1' to disable interrupt for event CLKOUTREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.9.6.28 INTPEND2

Address offset: 0x32C

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-L	R	COMPARE[i] (i=0..11)			Read pending status of interrupt for event COMPARE[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
M	R	RTCOMPARESYNC			Read pending status of interrupt for event RTCOMPARESYNC
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
N	R	PWMPERIODEND			Read pending status of interrupt for event PWMPERIODEND
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
O	R	PWMREADY			Read pending status of interrupt for event PWMREADY
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
P	R	CLKOUTREADY			Read pending status of interrupt for event CLKOUTREADY
			NotPending	0	Read: Not pending

Bit number	31	30	29	28	27	26	25	24	23	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
			Pending	1	Read: Pending

8.9.6.29 INTEN3

Address offset: 0x330

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-L	RW	COMPARE[i] (i=0..11)			Enable or disable interrupt for event COMPARE[i]
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	RTCOMPARESYNC			Enable or disable interrupt for event RTCOMPARESYNC
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	PWMPERIODEND			Enable or disable interrupt for event PWMPERIODEND
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	PWMREADY			Enable or disable interrupt for event PWMREADY
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	CLKOUTREADY			Enable or disable interrupt for event CLKOUTREADY
			Disabled	0	Disable
			Enabled	1	Enable

8.9.6.30 INTENSET3

Address offset: 0x334

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-L	RW W1S	COMPARE[i] (i=0..11)			Write '1' to enable interrupt for event COMPARE[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	RTCOMPARESYNC			Write '1' to enable interrupt for event RTCOMPARESYNC
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1S	PWMPERIODEND			Write '1' to enable interrupt for event PWMPERIODEND
			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				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
O	RW W1S	PWMREADY			Write '1' to enable interrupt for event PWMREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1S	CLKOUTREADY			Write '1' to enable interrupt for event CLKOUTREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.9.6.31 INTENCLR3

Address offset: 0x338

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-L	RW W1C	COMPARE[i] (i=0..11)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	RTCOMPARESYNC			Write '1' to disable interrupt for event RTCOMPARESYNC
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	PWMPERIODEND			Write '1' to disable interrupt for event PWMPERIODEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1C	PWMREADY			Write '1' to disable interrupt for event PWMREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1C	CLKOUTREADY			Write '1' to disable interrupt for event CLKOUTREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.9.6.32 INTPEND3

Address offset: 0x33C

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-L	R	COMPARE[i] (i=0..11)			Read pending status of interrupt for event COMPARE[i]																															
			NotPending	0	Read: Not pending																															
			Pending	1	Read: Pending																															
M	R	RTCOMPARESYNC			Read pending status of interrupt for event RTCOMPARESYNC																															
			NotPending	0	Read: Not pending																															
			Pending	1	Read: Pending																															
N	R	PWMPERIODEND			Read pending status of interrupt for event PWMPERIODEND																															
			NotPending	0	Read: Not pending																															
			Pending	1	Read: Pending																															
O	R	PWMREADY			Read pending status of interrupt for event PWMREADY																															
			NotPending	0	Read: Not pending																															
			Pending	1	Read: Pending																															
P	R	CLKOUTREADY			Read pending status of interrupt for event CLKOUTREADY																															
			NotPending	0	Read: Not pending																															
			Pending	1	Read: Pending																															

8.9.6.33 EVTEN

Address offset: 0x400

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									A																											
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value	Description																															
A	RW	PWMPERIODEND			Enable or disable event routing for event PWMPERIODEND																															
			Disabled	0	Disable																															
			Enabled	1	Enable																															

8.9.6.34 EVTENSET

Address offset: 0x404

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									A																											
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value	Description																															
A	RW	PWMPERIODEND			Write '1' to enable event routing for event PWMPERIODEND																															
			Disabled	0	Read: Disabled																															
			Enabled	1	Read: Enabled																															
			Set	1	Enable																															

8.9.6.35 EVTENCLR

Address offset: 0x408

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								A																											
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	PWMPERIODEND																																	
			Disabled	0																															
			Enabled	1																															
			Clear	1																															

8.9.6.36 MODE

Address offset: 0x510

Counter 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																																		B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	AUTOEN																																	
			Default	0																															
			CpuActive	1																															
B	RW	SYSCOUNTEREN																																	
			Disabled	0																															
			Enabled	1																															

8.9.6.37 CC[n].CCL (n=0..11)

Address offset: 0x520 + (n × 0x10)

The lower 32-bits of Capture/Compare register CC[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																															
A	RW	CCL																																	

8.9.6.38 CC[n].CCH (n=0..11)

Address offset: 0x524 + (n × 0x10)

The higher 32-bits of Capture/Compare register CC[n]

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CCH			Capture/Compare high value in 1 µs

8.9.6.39 CC[n].CCADD (n=0..11)

Address offset: 0x528 + (n × 0x10)

Count to add to CC[n] when this register 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 A A A A A 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	VALUE			Count to add to CC[n]
B	W	REFERENCE			Configure the Capture/Compare register
			SYSCOUNTER	0	Adds SYSCOUNTER value. CC[n] becomes CCADD[n].VALUE + SYSCOUNTER when this register is written.
			CC	1	Adds CC value. CC[n] becomes CCADD[n].VALUE + CC[n] when this register is written.

8.9.6.40 CC[n].CCEN (n=0..11)

Address offset: 0x52C + (n × 0x10)

Configure Capture/Compare register CC[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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	ACTIVE			Configure the Capture/Compare register
			Disable	0	Capture/Compare register CC[n] Disabled.
			Enable	1	Capture/Compare register CC[n] enabled.
B	RW	PASTCC			Status of event EVENTS_COMPARE[n] caused by the configured CC value is in past
			Inactive	0	Inactive
			Active	1	Active

8.9.6.41 TIMEOUT

Address offset: 0x6A4

Timeout after all CPUs gone into sleep state to stop the SYSCOUNTER

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A 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	VALUE			Number of 32Ki cycles

8.9.6.42 INTERVAL

Address offset: 0x6A8

Count to add to CC[0] when the event EVENTS_COMPARE[0] triggers.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	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	VALUE																		Count to add to CC[0]															

8.9.6.43 WAKETIME

Address offset: 0x6AC

GRTC wake up 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	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	VALUE																		Number of LFCLK clock cycles to wake up before the next scheduled EVENTS_COMPARE event															

8.9.6.44 STATUS.LFTIMER

Address offset: 0x6B0

Low frequency timer 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 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																		Low frequency timer is ready or busy.															
				Busy			0													Busy															
				Ready			1													Ready															

8.9.6.45 STATUS.PWM

Address offset: 0x6B4

PWM 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 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																		PWM is ready or busy.															
				Busy			0													Busy															
				Ready			1													Ready															

8.9.6.46 STATUS.CLKOUT

Address offset: 0x6B8

CLKOUT configuration 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 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	READY																																	
			Busy	0																															
			Ready	1																															

8.9.6.47 PWMCONFIG

Address offset: 0x710

PWM 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	COMPAREVALUE																																	

8.9.6.48 CLKOUT

Address offset: 0x714

Configuration of clock 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																																		B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	CLKOUT32K																																	
			Disabled	0																															
			Enabled	1																															
B	RW	CLKOUTFAST																																	
			Disabled	0																															
			Enabled	1																															

8.9.6.49 CLKCFG

Address offset: 0x718

Clock 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 A A A A A A
Reset 0x00010001				**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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	CLKFASTDIV		1..255	Fast clock divisor value of clock output Fast clock divisor value 0 behaves same as 1.
B	W	CLKSEL			GRTC LFCLK clock source selection
			LFXO	0	GRTC LFCLK clock source is LFXO
			SystemLFCLK	1	GRTC LFCLK clock source is system LFCLK
			LFLPRC	2	GRTC LFCLK clock source is LFLPRC

8.9.6.50 SYSCOUNTER[n].SYSCOUNTERL (n=0..3)

Address offset: 0x720 + (n × 0x10)

The lower 32-bits of the SYSCOUNTER for index [n]

The SYSCOUNTERL must be read before SYSCOUNTERH.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A 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	VALUE			The lower 32-bits of the SYSCOUNTER value.

8.9.6.51 SYSCOUNTER[n].SYSCOUNTERH (n=0..3)

Address offset: 0x724 + (n × 0x10)

The higher 20-bits of the SYSCOUNTER for index [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				D C B A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x60000000				**0 1 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	R	VALUE			The higher 20-bits of the SYSCOUNTER value.
B	R	LOADED			SYSCOUNTER loaded status
			NotLoaded	0	SYSCOUNTER is not loaded
			Loaded	1	SYSCOUNTER is loaded
C	R	BUSY			SYSCOUNTER busy status
			Ready	0	SYSCOUNTER is ready for read
			Busy	1	SYSCOUNTER is busy, so not ready for read (value returned in the VALUE field of this register is not valid)
D	R	OVERFLOW			The SYSCOUNTERL overflow indication after reading it.
			NoOverflow	0	SYSCOUNTERL is not overflown
			Overflow	1	SYSCOUNTERL overflown

8.9.6.52 SYSCOUNTER[n].ACTIVE (n=0..3)

Address offset: 0x728 + (n × 0x10)

Request to keep the SYSCOUNTER in the active state and prevent going to sleep for index [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																															
A	RW	ACTIVE																																	
			NotActive	0																															
			Active	1																															

8.10 PWM — Pulse width modulation

The pulse width modulation peripheral (PWM) enables the generation of pulse width modulated signals on GPIO. The peripheral implements a counter with up-count mode and up-and-down-count mode, consisting of four PWM channels that can drive assigned GPIO pins.

The main features of PWM are the following:

- 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 on every PWM period
- RAM sequences can be repeated or connected into loops

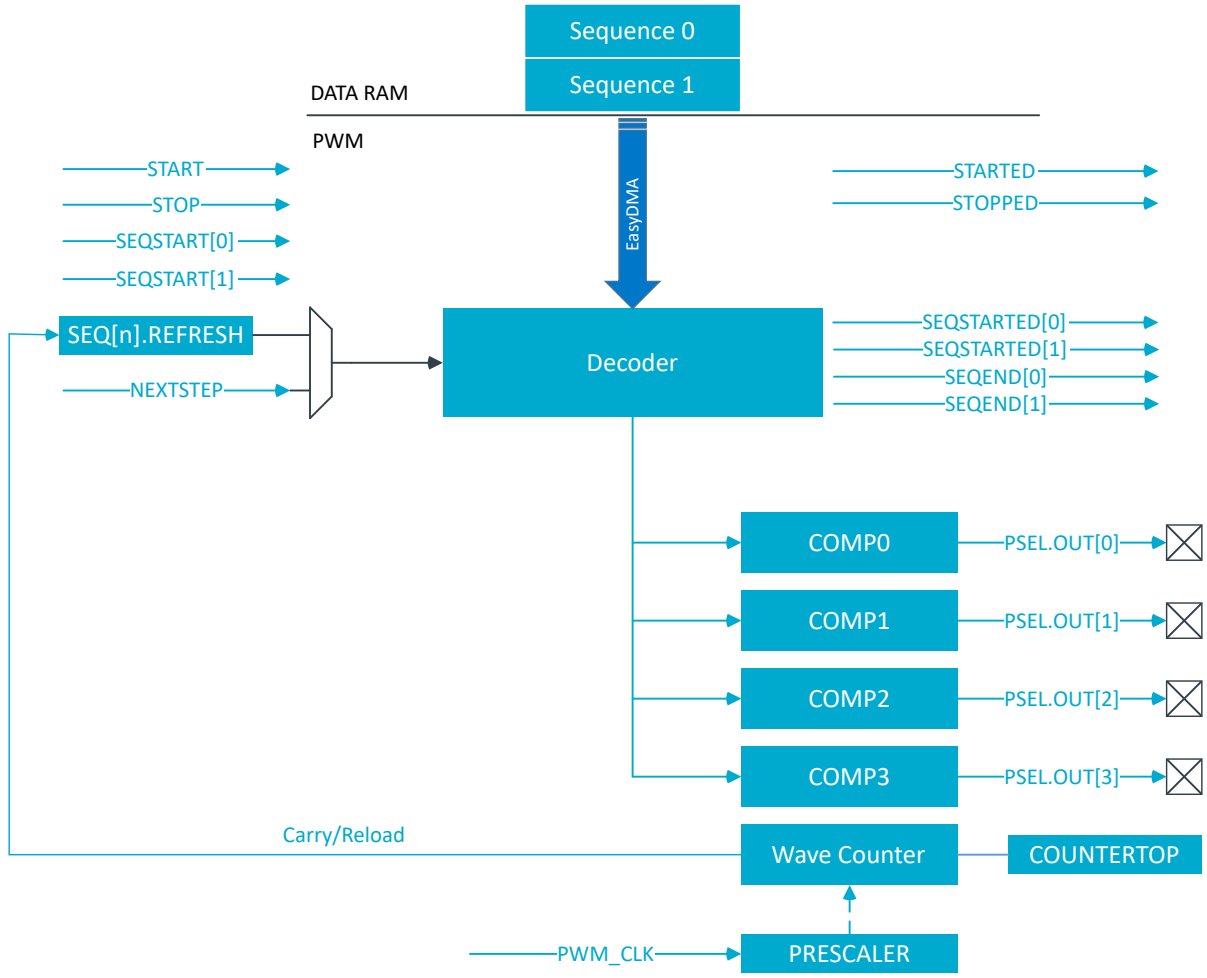


Figure 55: PWM module

8.10.1 Wave counter

The wave counter is responsible for generating the pulses at a duty cycle that depends on the compare values, and at a frequency that depends on COUNTERTOP.

There is one common 15-bit counter with four compare channels. Thus, all four channels will share the same period (PWM frequency), but can have individual duty cycle and polarity. The polarity is set by the most significant bit (MSB) of the value read from RAM (see figure Decoder memory access modes on page 231). When the MSB bit is high (FallingEdge polarity), OUT[n] starts high to become low during the given PWM cycle, whereas the inverse occurs for RisingEdge polarity. 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 231 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 two PWM channels with the same frequency but different duty cycle:

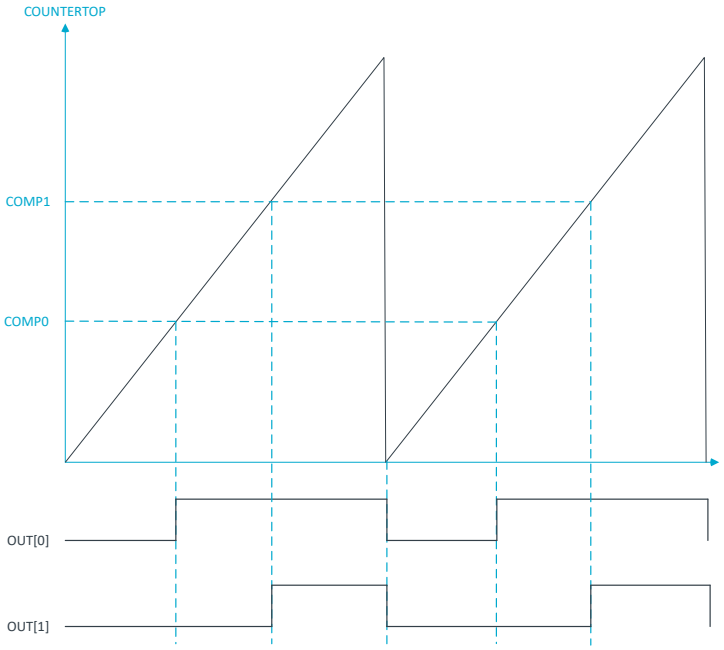


Figure 56: PWM counter in up mode example - RisingEdge polarity

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_port << PWM_PSEL_OUT_PORT_Pos) |
                        (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                                                PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_port << PWM_PSEL_OUT_PORT_Pos) |
                        (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->DMA.SEQ[0].PTR    = ((uint32_t)(pwm_seq) << PWM_DMA_SEQ_PTR_PTR_Pos);
NRF_PWM0->DMA.SEQ[0].MAXCNT = (sizeof(pwm_seq) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA.SEQ[0].START = 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:

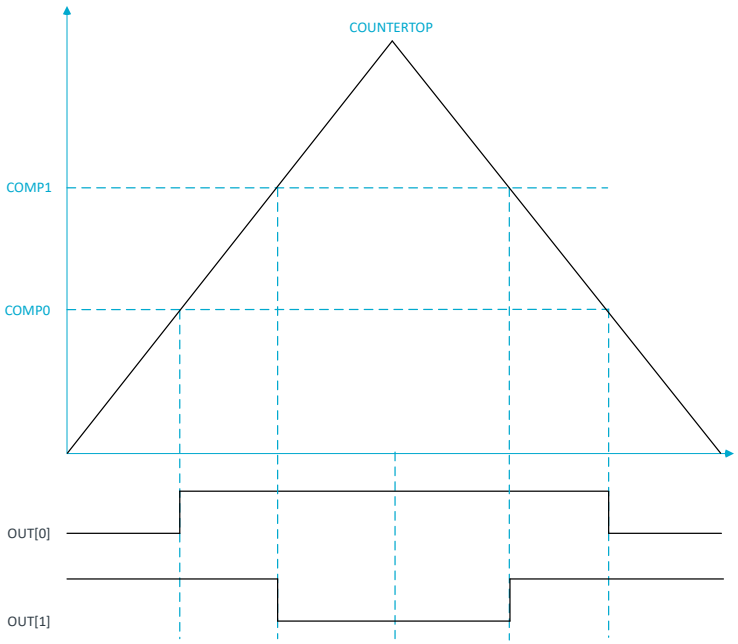


Figure 57: PWM counter in up-and-down mode example

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_port << PWM_PSEL_OUT_PORT_Pos) |
                        (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->DMA.SEQ[0].PTR    = ((uint32_t)(pwm_seq) << PWM_DMA_SEQ_PTR_PTR_Pos);
NRF_PWM0->DMA.SEQ[0].MAXCNT = (sizeof(pwm_seq) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA.SEQ[0].START = 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: $T_{steps} = T_{PWM\_CLK} * 2$

8.10.2 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:

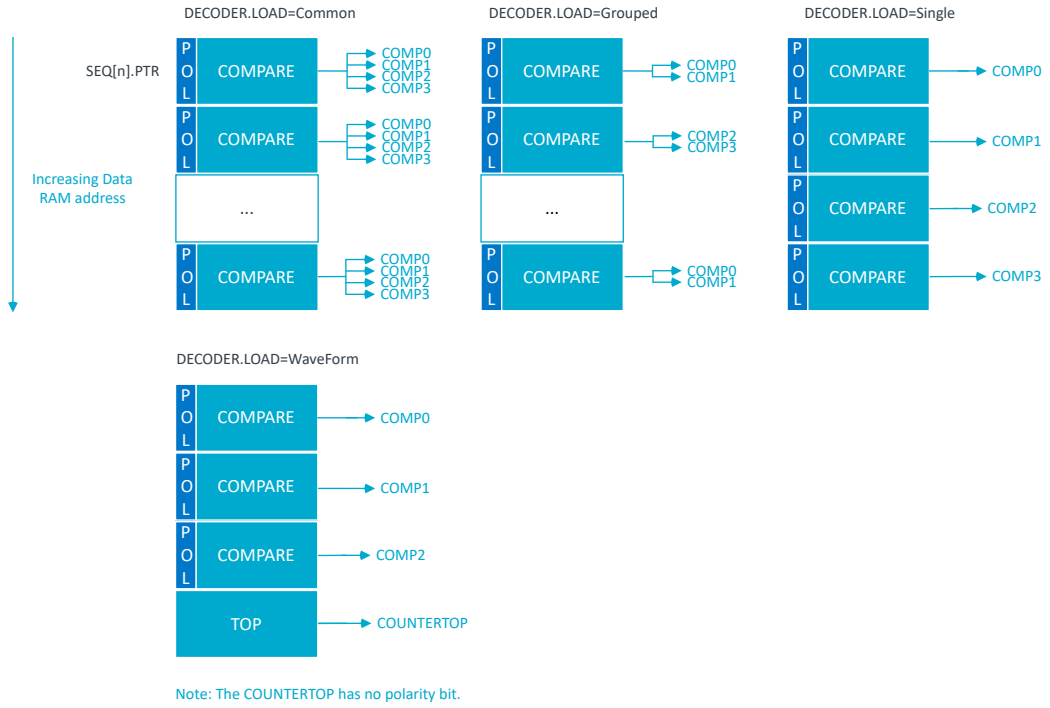


Figure 58: Decoder memory access modes

A special mode of operation is available when DECODER.LOAD is set to WaveForm. In WaveForm 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 11 for more information about the different memory regions. After the SEQ[n].PTR is set to the desired RAM location, the SEQ[n].MAXCNT register must be set to the number of bytes 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 DMA.SEQ[n].START 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.MAXCNT=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 output defined in the IDLEOUT register. The following figure illustrates an example of a simple playback.

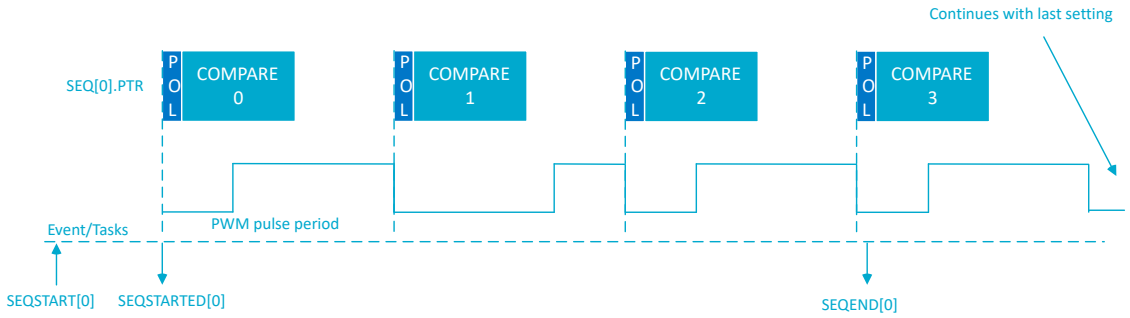


Figure 59: Simple sequence example

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_port << PWM_PSEL_OUT_PORT_Pos) |
                        (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->DMA.SEQ[0].PTR  = ((uint32_t)(seq0_ram) << PWM_DMA_SEQ_PTR_PTR_Pos);
NRF_PWM0->DMA.SEQ[0].MAXCNT  = (sizeof(seq0_ram) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA.SEQ[0].START = 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 by the IDLEOUT register. PWM generation can then only be restarted through a DMA.SEQ[n].START task. DMA.SEQ[n].START 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 DMA.SEQ[n].START task	After having received the SEQSTARTED[n] event
SEQ[n].MAXCNT	When sending the DMA.SEQ[n].START task	After having received the SEQSTARTED[n] event
SEQ[0].ENDDELAY	When sending the SEQSTART[0] task Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [0] through a SEQSTART[0] task When no more value from sequence [0] gets loaded from RAM (indicated by the SEQEND[0] event) At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].ENDDELAY	When sending the SEQSTART[1] task Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [1] through a SEQSTART[1] task When no more value from sequence [1] gets loaded from RAM (indicated by the SEQEND[1] event) At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
SEQ[0].REFRESH	When sending the SEQSTART[0] task Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [0] through a SEQSTART[0] task At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].REFRESH	When sending the SEQSTART[1] task Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	Before starting sequence [1] through a SEQSTART[1] task At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
COUNTERTOP	In DECODER.LOAD=WaveForm: this register is ignored. In all other LOAD modes: at the end of current PWM period (indicated by the PWMPERIODEND event)	Before starting PWM generation through a DMA.SEQ[n].START task After a STOP task has been triggered, and the STOPPED event has been received.
MODE	Immediately	Before starting PWM generation through a DMA.SEQ[n].START task After a STOP task has been triggered, and the STOPPED event has been received.
DECODER	Immediately	Before starting PWM generation through a DMA.SEQ[n].START task After a STOP task has been triggered, and the STOPPED event has been received.
PRESCALER	Immediately	Before starting PWM generation through a DMA.SEQ[n].START task After a STOP task has been triggered, and the STOPPED event has been received.
LOOP	Immediately	Before starting PWM generation through a DMA.SEQ[n].START 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 32: 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 256.

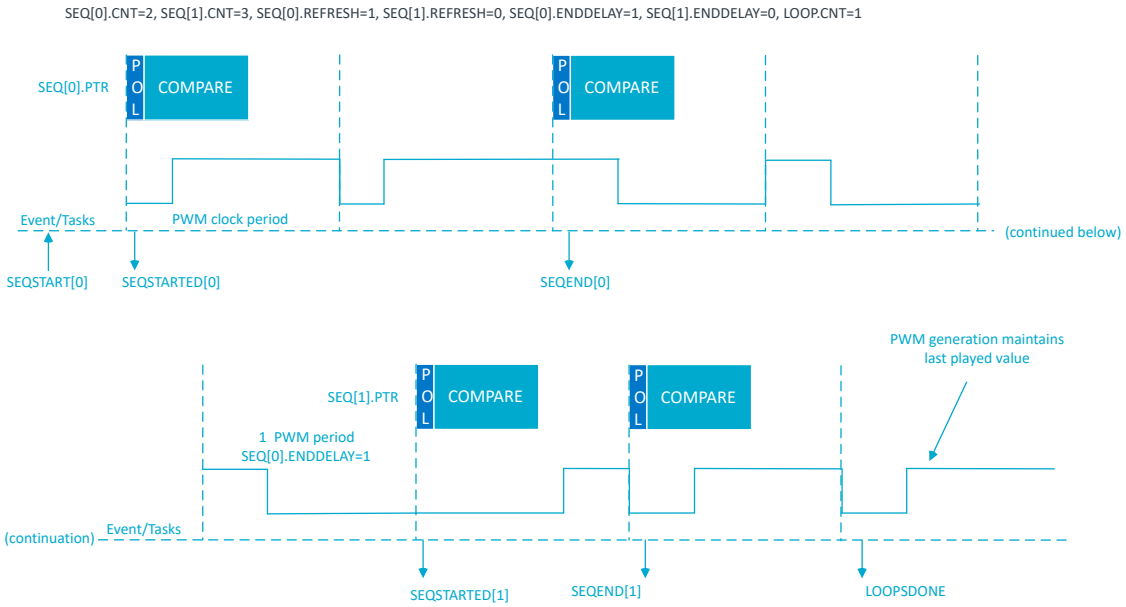


Figure 60: Example using two sequences

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].MAXCNT). 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_port << PWM_PSEL_OUT_PORT_Pos) |
                        (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->DMA.SEQ[0].PTR   = ((uint32_t)(seq0_ram) << PWM_DMA_SEQ_PTR_PTR_Pos);
NRF_PWM0->DMA.SEQ[0].MAXCNT  = (sizeof(seq0_ram) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[0].REFRESH  = 1;
NRF_PWM0->SEQ[0].ENDDELAY = 1;
NRF_PWM0->DMA.SEQ[1].PTR   = ((uint32_t)(seq1_ram) << PWM_DMA_SEQ_PTR_PTR_Pos);
NRF_PWM0->DMA.SEQ[1].MAXCNT  = (sizeof(seq1_ram) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[1].REFRESH  = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA.SEQ[0].START = 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))

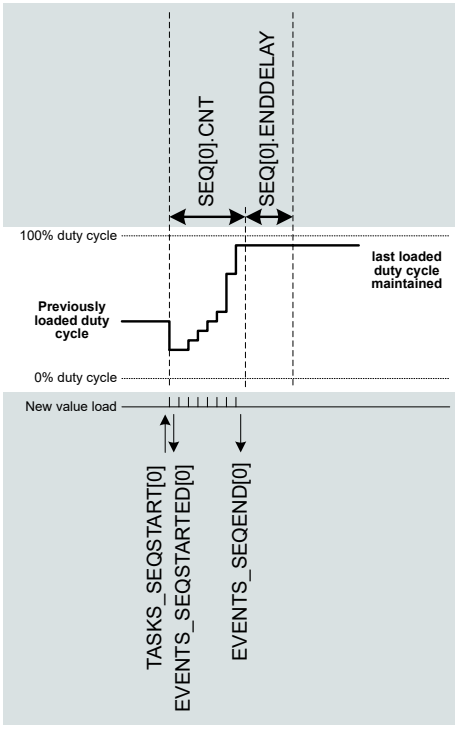


Figure 61: Single shot (LOOP.CNT=0)

Note: The single-shot example also applies to SEQ[1]. Only SEQ[0] is represented for simplicity.

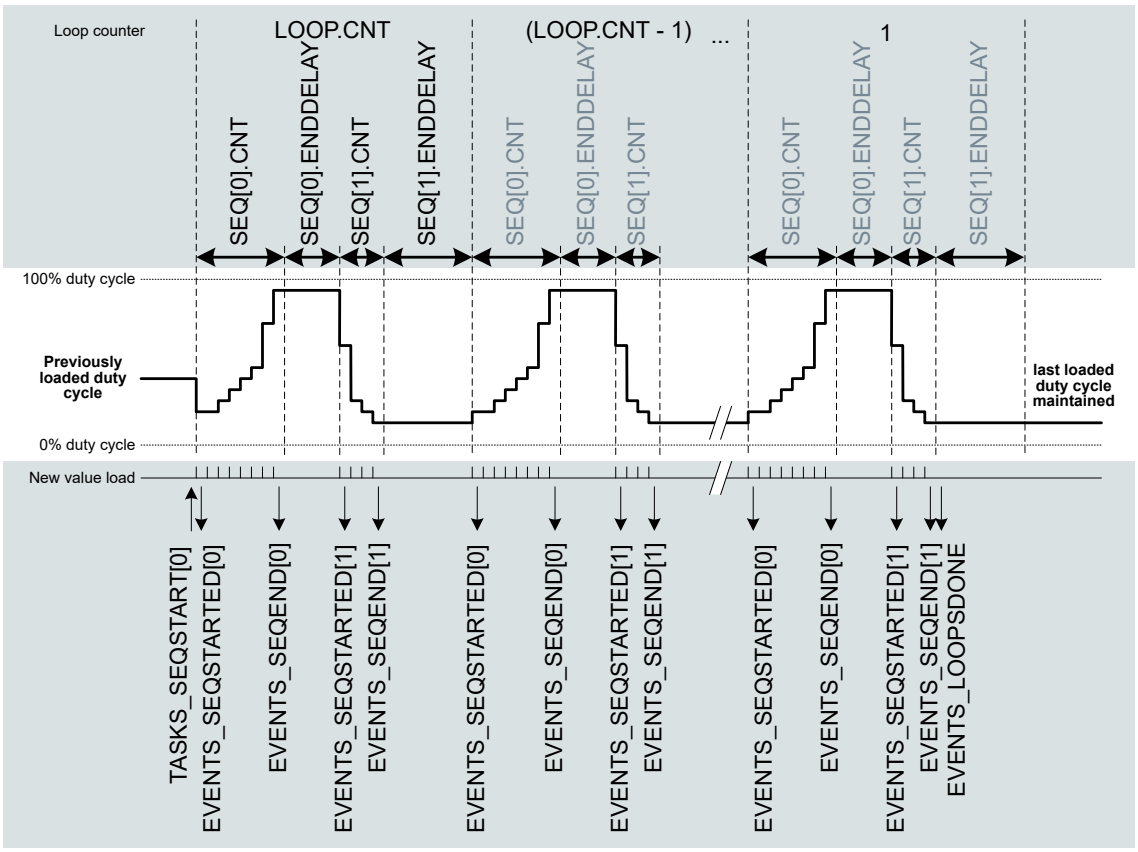


Figure 62: Complex sequence (LOOP.CNT>0) starting with SEQ[0]

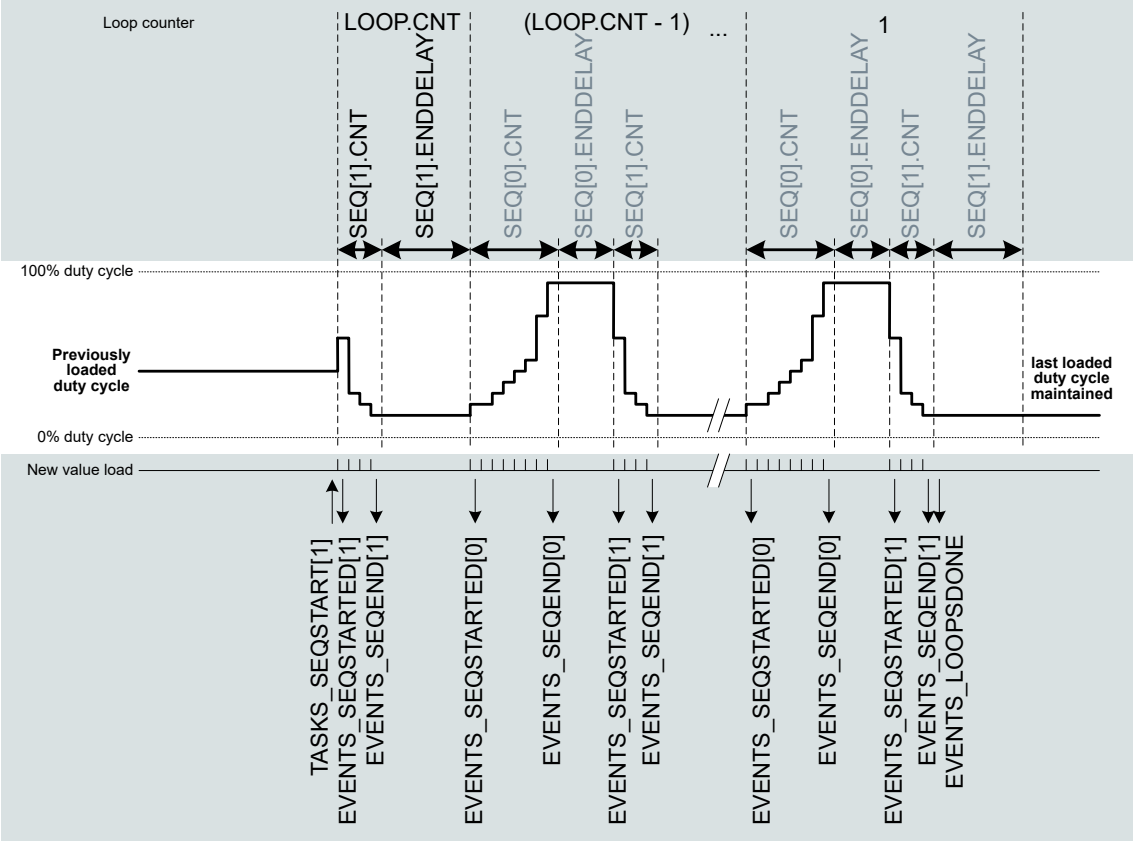


Figure 63: Complex sequence (LOOP.CNT>0) starting with SEQ[1]

Note: If a sequence is in use in a simple or complex sequence, it must have a length of SEQ[n].MAXCNT > 0.

This example shows how the PWM module can be configured to repeat a single sequence until stopped.

```
NRF_PWM0->PSEL.OUT[0] = (first_port << PWM_PSEL_OUT_PORT_Pos) |
                        (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 DMA.SEQ1.START task for infinite loop
NRF_PWM0->SHORTS      = (PWM_SHORTS_LOOPSDONE_DMA_SEQ1_START_Enabled <<
                                        PWM_SHORTS_LOOPSDONE_DMA_SEQ1_START_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->DMA.SEQ[1].PTR  = ((uint32_t)(seq0_ram)) << PWM_DMA_SEQ_PTR_PTR_Pos;
NRF_PWM0->DMA.SEQ[1].MAXCNT  =(sizeof(seq0_ram) << PWM_DMA_SEQ_MAXCNT_MAXCNT_Pos);
NRF_PWM0->SEQ[1].REFRESH  = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_DMA.SEQ[1].START = 1;
```

8.10.3 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].MAXCNT > 0 when sequence playback starts.

8.10.4 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.

Once PWM has been enabled, the PSEL.OUT[n] registers take effect and PWM generation starts from the IDLEOUT register. PWM can then be started and sequences generated.

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 and the IDLEOUT register

Table 33: Recommended GPIO configuration before starting PWM generation

The idle state of a pin is defined by the OUT register in the GPIO module and the IDLEOUT register, to ensure that the pins used by the PWM module are driven correctly. Both OUT register in the GPIO module and the IDLEOUT register should be set with same value for each PWM channel before enabling the PWM module . When PWM is disabled using the ENABLE register the PWM module stops controlling the GPIO pins, and the corresponding pins are then controlled by the GPIO peripheral.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

8.10.5 Registers

Instances

Instance	Domain	Base address	Description
PWM20	GLOBAL	0x400D2000	Pulse width modulation unit PWM20

Configuration

Instance	Domain	Configuration
PWM20	GLOBAL	Use GPIO port P1 IDLEOUT register is available. EVENTS_COMPAREMATCH events are available. CURRENTAMOUNT register included.

Register overview

Register	Offset	Description
TASKS_STOP	0x004	Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback
TASKS_NEXTSTEP	0x008	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.
TASKS_DMA.SEQ[n].START	0x010	Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
TASKS_DMA.SEQ[n].STOP	0x014	Stops operation using easyDMA. This does not trigger an END event.
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
SUBSCRIBE_NEXTSTEP	0x088	Subscribe configuration for task NEXTSTEP
SUBSCRIBE_DMA.SEQ[n].START	0x090	Subscribe configuration for task START
SUBSCRIBE_DMA.SEQ[n].STOP	0x094	Subscribe configuration for task STOP
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
EVENTS_RAMUNDERFLOW	0x120	Emitted when retrieving from RAM does not complete in time for the PWM module
EVENTS_DMA.SEQ[n].END	0x124	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.SEQ[n].READY	0x128	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.SEQ[n].BUSERROR	0x12C	An error occured during the bus transfer.
EVENTS_COMPAREMATCH[n]	0x13C	This event is generated when the compare matches for the compare channel [n].
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
PUBLISH_RAMUNDERFLOW	0x1A0	Publish configuration for event RAMUNDERFLOW
PUBLISH_DMA.SEQ[n].END	0x1A4	Publish configuration for event END
PUBLISH_DMA.SEQ[n].READY	0x1A8	Publish configuration for event READY
PUBLISH_DMA.SEQ[n].BUSERROR	0x1AC	Publish configuration for event BUSERROR
PUBLISH_COMPAREMATCH[n]	0x1BC	Publish configuration for event COMPAREMATCH[n]
SHORTS	0x200	Shortcuts between local events and tasks
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
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
IDLEOUT	0x518	Configure the output value on the PWM channel during idle
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
DMA.SEQ[n].PTR	0x704	RAM buffer start address
DMA.SEQ[n].MAXCNT	0x708	Maximum number of bytes in channel buffer
DMA.SEQ[n].AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event.
DMA.SEQ[n].CURRENTAMOUNT	0x710	Number of bytes transferred in the current transaction

Register	Offset	Description
DMA.SEQ[n].TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.SEQ[n].BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.

8.10.5.1 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

8.10.5.2 TASKS_NEXTSTEP

Address offset: 0x008

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

8.10.5.3 TASKS_DMA

Peripheral tasks.

8.10.5.3.1 TASKS_DMA.SEQ[n] (n=0..1)

Peripheral tasks.

8.10.5.3.1.1 TASKS_DMA.SEQ[n].START (n=0..1)

Address offset: 0x010 + (n × 0x8)

Starts operation using easyDMA to load the values. See peripheral description for operation using 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	W	START			Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
			Trigger	1	Trigger task

8.10.5.3.1.2 TASKS_DMA.SEQ[n].STOP (n=0..1)

Address offset: 0x014 + (n × 0x8)

Stops operation using easyDMA. This does not trigger an END 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	STOP			Stops operation using easyDMA. This does not trigger an END event.
			Trigger	1	Trigger task

8.10.5.4 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.10.5.5 SUBSCRIBE_NEXTSTEP

Address offset: 0x088

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task NEXTSTEP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.10.5.6 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.10.5.6.1 SUBSCRIBE_DMA.SEQ[n] (n=0..1)

Subscribe configuration for tasks

8.10.5.6.1.1 SUBSCRIBE_DMA.SEQ[n].START (n=0..1)

Address offset: 0x090 + (n × 0x8)

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.10.5.6.1.2 SUBSCRIBE_DMA.SEQ[n].STOP (n=0..1)

Address offset: 0x094 + (n × 0x8)

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.10.5.7 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

8.10.5.8 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

8.10.5.9 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

8.10.5.10 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

8.10.5.11 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

8.10.5.12 EVENTS_RAMUNDERFLOW

Address offset: 0x120

Emitted when retrieving from RAM does not complete in time for the PWM 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	EVENTS_RAMUNDERFLOW			Emitted when retrieving from RAM does not complete in time for the PWM module
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.10.5.13 EVENTS_DMA

Peripheral events.

8.10.5.13.1 EVENTS_DMA.SEQ[n] (n=0..1)

Peripheral events.

8.10.5.13.1.1 EVENTS_DMA.SEQ[n].END (n=0..1)

Address offset: 0x124 + (n × 0xC)

Generated after all MAXCNT bytes have been transferred

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated after all MAXCNT bytes have been transferred
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.10.5.13.1.2 EVENTS_DMA.SEQ[n].READY (n=0..1)

Address offset: 0x128 + (n × 0xC)

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.10.5.13.1.3 EVENTS_DMA.SEQ[n].BUSERROR (n=0..1)

Address offset: 0x12C + (n × 0xC)

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.10.5.14 EVENTS_COMPAREMATCH[n] (n=0..3)

Address offset: 0x13C + (n × 0x4)

This event is generated when the compare matches for the 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	EVENTS_COMPAREMATCH			This event is generated when the compare matches for the compare channel [n].
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.10.5.15 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		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.16 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		[0..255]	DPPI channel that event SEQSTARTED[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.17 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event SEQEND[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.18 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event PWMPERIODEND will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.19 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		[0..255]	DPPI channel that event LOOPSDONE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.20 PUBLISH_RAMUNDERFLOW

Address offset: 0x1A0

Publish configuration for event RAMUNDERFLOW

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event RAMUNDERFLOW will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.21 PUBLISH_DMA

Publish configuration for events

8.10.5.21.1 PUBLISH_DMA.SEQ[n] (n=0..1)

Publish configuration for events

8.10.5.21.1.1 PUBLISH_DMA.SEQ[n].END (n=0..1)

Address offset: 0x1A4 + (n × 0xC)

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.21.1.2 PUBLISH_DMA.SEQ[n].READY (n=0..1)

Address offset: 0x1A8 + (n × 0xC)

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.21.1.3 PUBLISH_DMA.SEQ[n].BUSERROR (n=0..1)

Address offset: 0x1AC + (n × 0xC)

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.22 PUBLISH_COMPAREMATCH[n] (n=0..3)

Address offset: 0x1BC + (n × 0x4)

Publish configuration for event COMPAREMATCH[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		[0..255]	DPPI channel that event COMPAREMATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.10.5.23 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				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-B	RW	SEQEND[i]_STOP (i=0..1)			Shortcut between event SEQEND[n] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C-D	RW	LOOPSDONE_DMA_SEQ[i]_START (i=0..1)			Shortcut between event LOOPSDONE and task DMA.SEQ[n].START
			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
F	RW	RAMUNDERFLOW_STOP			Shortcut between event RAMUNDERFLOW and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G-H	RW	DMA_SEQ[i]_BUSERROR_STOP (i=0..1)			Shortcut between event DMA.SEQ[n].BUSERROR and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.10.5.24 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				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	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
B-C	RW	SEQSTARTED[i] (i=0..1)			Enable or disable interrupt for event SEQSTARTED[i]
			Disabled	0	Disable
			Enabled	1	Enable
D-E	RW	SEQEND[i] (i=0..1)			Enable or disable interrupt for event SEQEND[i]
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	PWMPERIODEND			Enable or disable interrupt for event PWMPERIODEND
			Disabled	0	Disable
			Enabled	1	Enable
G	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
H	RW	RAMUNDERFLOW			Enable or disable interrupt for event RAMUNDERFLOW
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	DMASEQ0END			Enable or disable interrupt for event DMASEQ0END
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	DMASEQ0READY			Enable or disable interrupt for event DMASEQ0READY
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	DMASEQ0BUSERROR			Enable or disable interrupt for event DMASEQ0BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	DMASEQ1END			Enable or disable interrupt for event DMASEQ1END
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	DMASEQ1READY			Enable or disable interrupt for event DMASEQ1READY
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	DMASEQ1BUSERROR			Enable or disable interrupt for event DMASEQ1BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable
O-R	RW	COMPAREMATCH[i] (i=0..3)			Enable or disable interrupt for event COMPAREMATCH[i]
			Disabled	0	Disable
			Enabled	1	Enable

8.10.5.25 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														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 W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B-C	RW W1S	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
D-E	RW W1S	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
F	RW W1S	PWMPERIODEND			Write '1' to enable interrupt for event PWMPERIODEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	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
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1S	RAMUNDERFLOW			Write '1' to enable interrupt for event RAMUNDERFLOW
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1S	DMASEQ0END			Write '1' to enable interrupt for event DMASEQ0END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1S	DMASEQ0READY			Write '1' to enable interrupt for event DMASEQ0READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1S	DMASEQ0BUSERROR			Write '1' to enable interrupt for event DMASEQ0BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			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				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 W1S	DMASEQ1END			Write '1' to enable interrupt for event DMASEQ1END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	DMASEQ1READY			Write '1' to enable interrupt for event DMASEQ1READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1S	DMASEQ1BUSERROR			Write '1' to enable interrupt for event DMASEQ1BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O-R	RW W1S	COMPAREMATCH[i] (i=0..3)			Write '1' to enable interrupt for event COMPAREMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.10.5.26 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				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 W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B-C	RW W1C	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
D-E	RW W1C	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
F	RW W1C	PWMPERIODEND			Write '1' to disable interrupt for event PWMPERIODEND
			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				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
G	RW W1C	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
H	RW W1C	RAMUNDERFLOW			Write '1' to disable interrupt for event RAMUNDERFLOW
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1C	DMASEQ0END			Write '1' to disable interrupt for event DMASEQ0END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1C	DMASEQ0READY			Write '1' to disable interrupt for event DMASEQ0READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1C	DMASEQ0BUSERROR			Write '1' to disable interrupt for event DMASEQ0BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	DMASEQ1END			Write '1' to disable interrupt for event DMASEQ1END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	DMASEQ1READY			Write '1' to disable interrupt for event DMASEQ1READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	DMASEQ1BUSERROR			Write '1' to disable interrupt for event DMASEQ1BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O-R	RW W1C	COMPAREMATCH[i] (i=0..3)			Write '1' to disable interrupt for event COMPAREMATCH[i]
			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				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

8.10.5.27 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				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	R	STOPPED			Read pending status of interrupt for event STOPPED
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
B-C	R	SEQSTARTED[i] (i=0..1)			Read pending status of interrupt for event SEQSTARTED[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
D-E	R	SEQEND[i] (i=0..1)			Read pending status of interrupt for event SEQEND[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
F	R	PWMPERIODEND			Read pending status of interrupt for event PWMPERIODEND
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
G	R	LOOPSDONE			Read pending status of 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.
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
H	R	RAMUNDERFLOW			Read pending status of interrupt for event RAMUNDERFLOW
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
I	R	DMASEQ0END			Read pending status of interrupt for event DMASEQ0END
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
J	R	DMASEQ0READY			Read pending status of interrupt for event DMASEQ0READY
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
K	R	DMASEQ0BUSERROR			Read pending status of interrupt for event DMASEQ0BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
L	R	DMASEQ1END			Read pending status of interrupt for event DMASEQ1END
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
M	R	DMASEQ1READY			Read pending status of interrupt for event DMASEQ1READY
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
N	R	DMASEQ1BUSERROR			Read pending status of interrupt for event DMASEQ1BUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending
O-R	R	COMPAREMATCH[i] (i=0..3)			Read pending status of interrupt for event COMPAREMATCH[i]
			NotPending	0	Read: Not pending
			Pending	1	Read: Pending

8.10.5.28 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

8.10.5.29 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

8.10.5.30 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.

8.10.5.31 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)

8.10.5.32 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

8.10.5.33 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)

8.10.5.34 IDLEOUT

Address offset: 0x518

Configure the output value on the PWM channel during idle

Writes to this register are ignored when the PWM 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 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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-D	RW	VAL[i] (i=0..3)																																	
				Idle output value for PWM channel [i]																															

8.10.5.35 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
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	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																															

8.10.5.36 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	CNT																																	
				Time added after the sequence in PWM periods																															

8.10.5.37 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	B	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																															
A	RW	PIN		[0..31]																															
				Pin number																															
B	RW	PORT		[0..7]																															
				Port number																															
C	RW	CONNECT																																	
				Connection																															
			Disconnected	1																															
				Disconnect																															
			Connected	0																															
				Connect																															

8.10.5.38 DMA.SEQ[n].PTR (n=0..1)

Address offset: 0x704 + (n × 0x24)

RAM buffer 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			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.																																	

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.10.5.39 DMA.SEQ[n].MAXCNT (n=0..1)

Address offset: 0x708 + (n × 0x24)

Maximum number of bytes in channel 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																																						
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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..0x7fff]	Maximum number of bytes in channel buffer																																	

8.10.5.40 DMA.SEQ[n].AMOUNT (n=0..1)

Address offset: 0x70C + (n × 0x24)

Number of bytes transferred in the last transaction, updated after the END 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																																						
Reset 0x00000000					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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..0x7fff]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.																																	

8.10.5.41 DMA.SEQ[n].CURRENTAMOUNT (n=0..1)

Address offset: 0x710 + (n × 0x24)

Number of bytes transferred in the current 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		[1..0x7fff]	Number of bytes transferred in the current transaction. Continuously updated.																																	

8.10.5.42 DMA.SEQ[n].TERMINATEONBUSERROR (n=0..1)

Address offset: 0x71C + (n × 0x24)

Terminate the transaction if a BUSERROR event is 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	ENABLE																																	
			Disabled	0								Disable																							
			Enabled	1								Enable																							

8.10.5.43 DMA.SEQ[n].BUSERRORADDRESS (n=0..1)

Address offset: 0x720 + (n × 0x24)

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	ADDRESS																																	

8.11 QDEC — Quadrature decoder

The Quadrature decoder (QDEC) provides buffered decoding of quadrature-encoded sensor signals. It is suitable for mechanical and optical sensors.

The main features of QDEC are:

- Digital waveform decoding from off-chip quadrature encoder.
- Sample accumulation eliminating hard real-time requirements to be enforced on application.
- Configurable sample period and accumulation to match application requirements.
- Optional input de-bounce filters.
- Optional LED output signal for optical encoders.

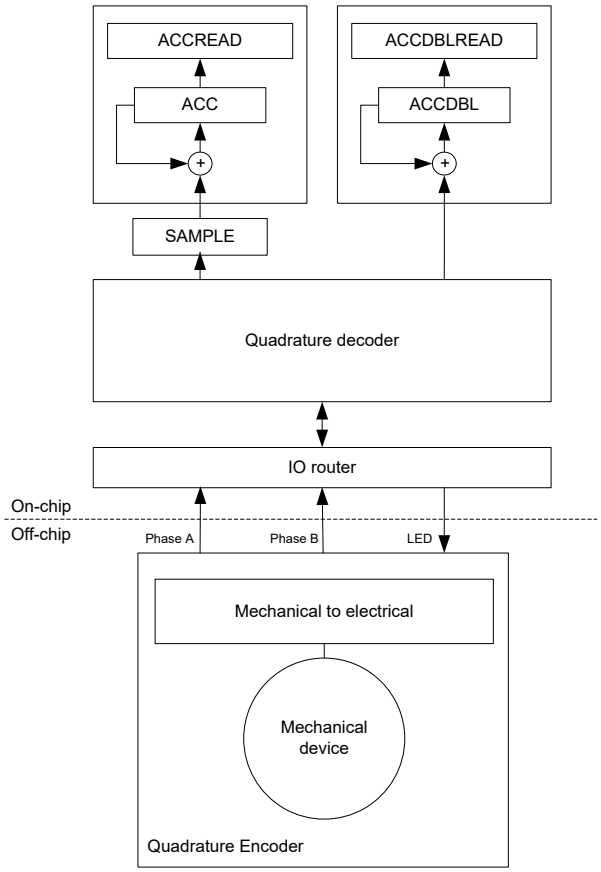


Figure 64: Quadrature decoder configuration

8.11.1 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 34: Sampled value encoding

8.11.2 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.

8.11.3 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.

8.11.4 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.

8.11.5 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.

8.11.6 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 263 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 35: GPIO configuration before enabling peripheral

8.11.7 Registers

Instances

Instance	Domain	Base address	Description
QDEC20	GLOBAL	0x400E0000	Quadrature decoder QDEC20

Configuration

Instance	Domain	Configuration
QDEC20	GLOBAL	Use GPIO port P1

Register overview

Register	Offset	Description
TASKS_START	0x000	Task starting the quadrature decoder
TASKS_STOP	0x004	Task stopping the quadrature decoder
TASKS_READCLRACC	0x008	Read and clear ACC and ACCDBL
TASKS_RDCLRACC	0x00C	Read and clear ACC
TASKS_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
SAMPLEPER	0x508	Sample period

Register	Offset	Description
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

8.11.7.1 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

8.11.7.2 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

8.11.7.3 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

8.11.7.4 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

8.11.7.5 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

8.11.7.6 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.11.7.7 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.11.7.8 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task READCLRACC will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.11.7.9 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RDCLRACC will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.11.7.10 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RDCLRDBL will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.11.7.11 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

8.11.7.12 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

8.11.7.13 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

8.11.7.14 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

8.11.7.15 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

8.11.7.16 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		[0..255]	DPPI channel that event SAMPLERDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.11.7.17 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		[0..255]	DPPI channel that event REPORTRDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.11.7.18 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event ACCOF will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.11.7.19 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DBLRDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.11.7.20 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.11.7.21 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
			Disabled	0	Disable shortcut

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

8.11.7.22 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 W1S	SAMPLERDY			Write '1' to enable interrupt for event SAMPLERDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	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 W1S	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 W1S	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 W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.11.7.23 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 W1C	SAMPLERDY			Write '1' to disable interrupt for event SAMPLERDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	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 W1C	ACCOF			Write '1' to disable interrupt for event ACCOF
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	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 W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.11.7.24 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

8.11.7.25 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

8.11.7.26 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
			256us	1	256 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
			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

8.11.7.27 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.

8.11.7.28 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

8.11.7.29 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.

8.11.7.30 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.

8.11.7.31 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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.11.7.32 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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.11.7.33 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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.11.7.34 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

8.11.7.35 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

8.11.7.36 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																															
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.																																

8.11.7.37 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																															
A	R	ACCDBLREAD		[0..15]																															
				Snapshot of the ACCDBL register. This field is updated when the READCLRACC or RDCLRDBL task is triggered.																																

8.12 RADIO — 2.4 GHz radio

The 2.4 GHz radio transceiver is compatible with multiple radio standards such as Bluetooth Low Energy, and Nordic's proprietary protocols.

The main features of the RADIO peripheral are the following:

- Multidomain 2.4 GHz radio transceiver, with
  - Bluetooth Low Energy 1 Mbps and 2 Mbps modes
  - 1 Mbps, 2 Mbps and 4 Mbps Nordic proprietary modes
- Efficient data interface with EasyDMA support
- Automatic address filtering and pattern matching
- Automated packet assembler/disassembler
- Automated CRC generator and checker

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.

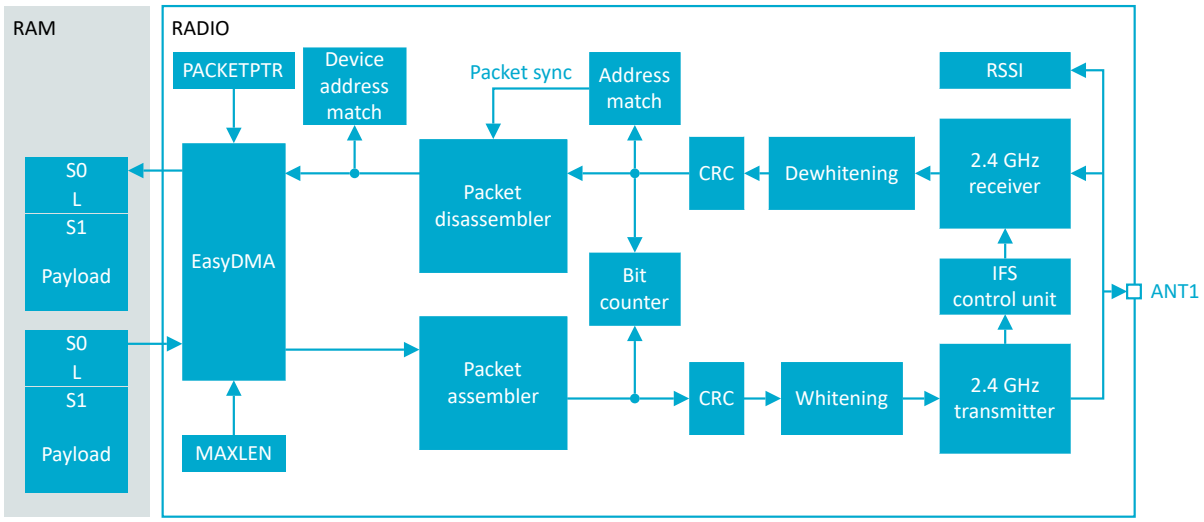


Figure 65: RADIO block diagram

RADIO includes a device address match unit and an interframe spacing control unit that can be utilized to simplify device filtering 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.

8.12.1 Packet configuration

A RADIO packet contains the fields PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD, and CRC.

The content of a RADIO packet is illustrated in the following figure. RADIO sends the fields in the packet according to the sequence shown in the figure, starting on the left.

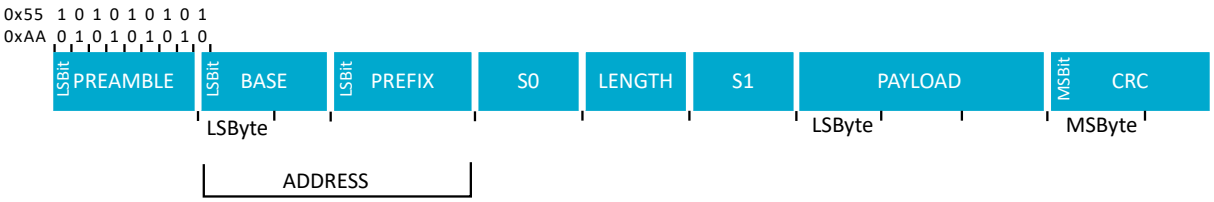


Figure 66: On-air packet layout

Not shown in the figure 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, and is configured using PCNF0.PLEN.

		Preamble	
MODE	Preamble size (octets)	When first bit of ADDRESS is 0	When first bit of ADDRESS is 1
Ble_1Mbit	1	0xAA	0x55
Nrf_1Mbit	1	0xAA	0x55
Nrf_2Mbit	1	0xAA	0x55
Ble_2Mbit	2	0xAAAA	0x5555
Nrf_4Mbit_0BT4	2	0xAAAA	0x5555
Nrf_4Mbit_0BT6	2	0xAAAA	0x5555

Table 36: Preamble size according to mode

Radio packets are stored in memory inside instances of a RADIO packet data structure as shown in the following figure. The PREAMBLE, ADDRESS and CRC fields are omitted in this data structure. Fields S0, LENGTH, and S1 are optional.



Figure 67: 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, meaning 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.

8.12.2 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 the PCNF1.BALEN register. The base address is truncated from the least significant byte if PCNF1.BALEN is less than 4. See Definition of logical addresses on page 280.

The on-air addresses are defined in the BASE0/BASE1 registers 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 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 37: Definition of logical addresses

8.12.3 Data whitening

Packet whitening and de-whitening is possible with RADIO and is enabled in PCNF1.WHITEEN. When enabled, whitening and de-whitening will be handled by RADIO automatically as packets are sent and received.

The data whitening is done by means of a configurable linear feedback shift register in a one-to-many topology, as illustrated in the following figure. Bit 0 is used to exclusive OR (XOR) the data packet that is to be whitened or de-whitened. The linear feedback shift register is configured and initialized using the DATAWHITE register. The reset value for the DATAWHITE.POLY field is compatible with Bluetooth Low Energy. The initial vector is configured in DATAWHITE.IV.

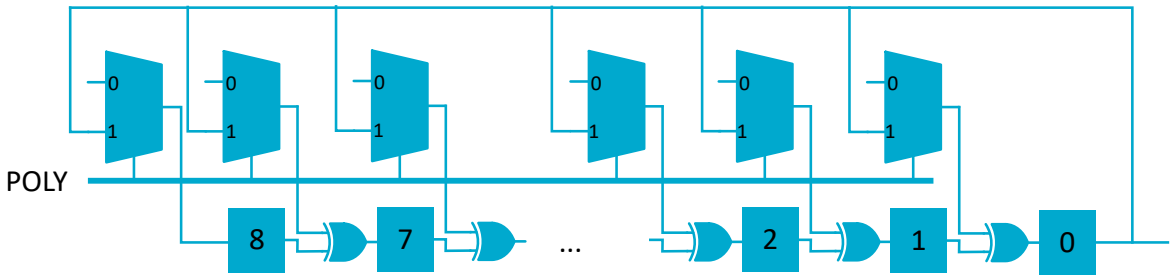


Figure 68: Data whitening and de-whitening

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.

8.12.4 CRC

The cyclic redundancy check (CRC) generator in RADIO calculates the CRC over the whole packet excluding the preamble.

The device also supports excluding the address field from the CRC calculation, see the CRCCNF.SKIPADDR register for more information.

The CRC polynomial is configurable as illustrated in the following figure, where bit `0` in the register CRCPOLY corresponds to X^0 and bit `n` corresponds to X^n. See CRCPOLY on page 325 for more information.

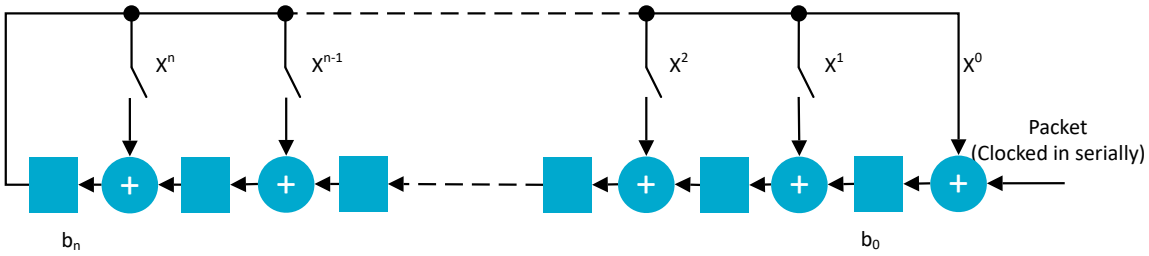


Figure 69: n bit CRC generation

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 $b_{[0..n]}$ will be initialized with a predefined value specified in the register CRCINIT. After the whole packet has been clocked through the CRC generator, $b_{[0..n]}$ will hold the resulting CRC. This value will be used by RADIO during both transmission and reception. The CPU cannont read latches $b_{[0..n]}$ at any time. However, a received CRC can be read by the CPU via the register RXCRC.

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

When the entire packet including the CRC has been received, and no errors were detected, RADIO generates the event CRCOK. If CRC errors were detected, the event CRCERROR is generated.

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

8.12.5 Radio states

Tasks and events are used to control the operating state of RADIO when in RX mode or TX mode.

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
RXDISABLE	RADIO is disabling the receiver
SETTLE	RADIO is waiting for the PLL to settle
PLL	The PLL has settled
TXRU	RADIO is ramping up and preparing for transmission
TXIDLE	RADIO is ready for transmission to start
TX	RADIO is transmitting a packet
TXDISABLE	RADIO is disabling the transmitter

Table 38: RADIO state diagram

A simplified state diagram without the optional states SETTLE and PLL is shown in the following figure.

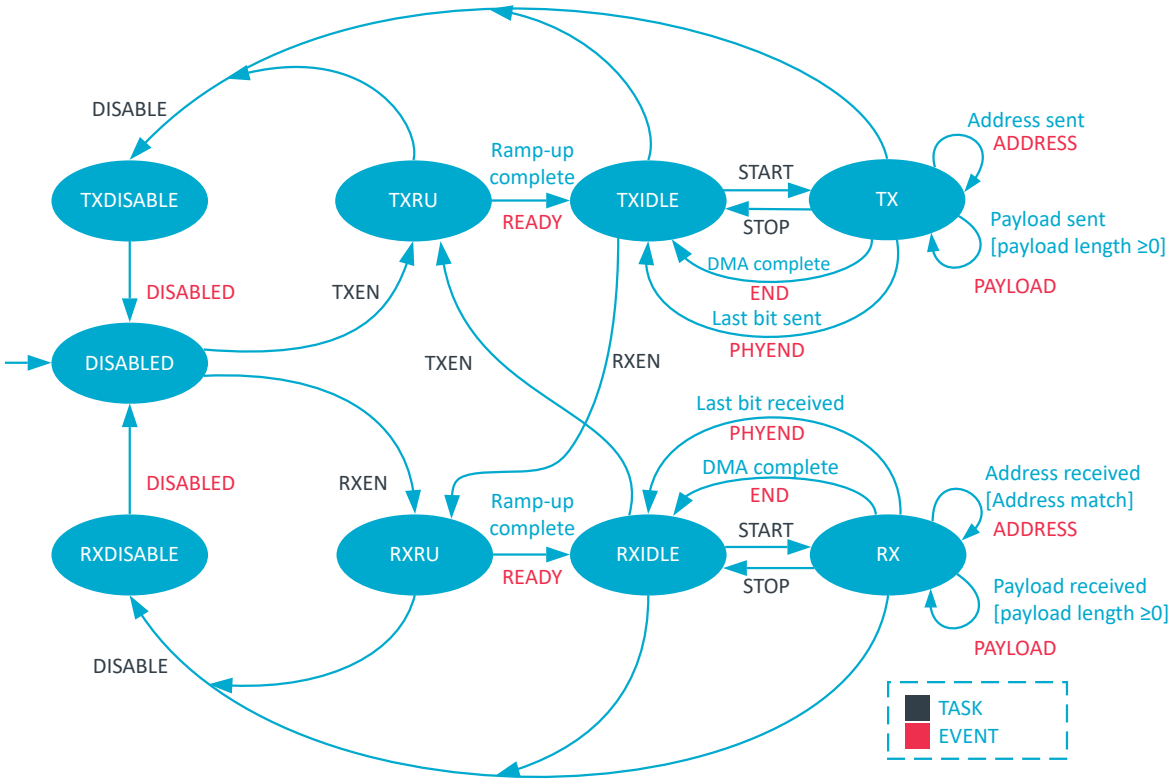
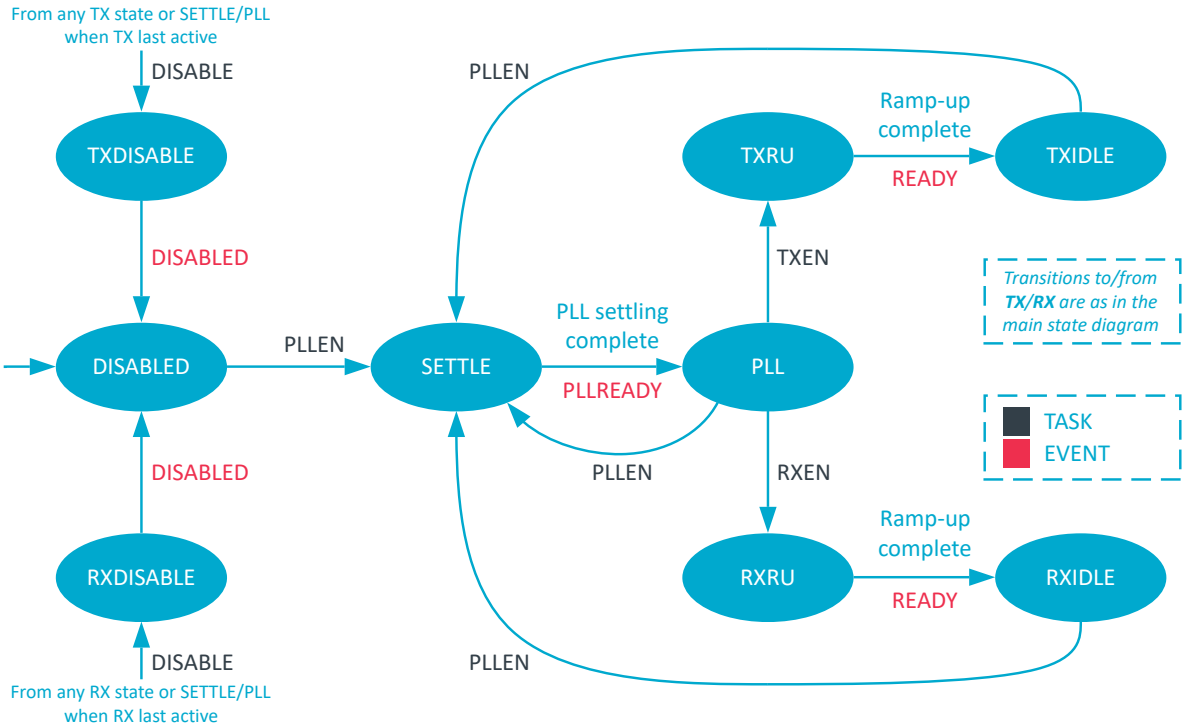


Figure 70: Radio states without PLL

A second state diagram is shown in the following figure. This diagram only contains the specific transitions involving SETTLE and PLL. This way of operating the RADIO enables keeping the PLL on and locked without enabling RX or TX.

Max 700 px



Max 750 px

Figure 71: Radio states, PLL only

This figure shows how the tasks and events relate to the RADIO peripheral's operation. RADIO does not prevent a task from being triggered from the wrong state. For example, if the RXEN task is triggered from the RXDISABLE state, this may lead to unpredictable behavior. The PAYLOAD event is always generated, even if the payload is zero.

8.12.6 Transmit sequence

Before RADIO can transmit a packet, it must first ramp-up in TX mode. See TXRU in Radio states without PLL on page 282 and Transmit sequence on page 283. 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, meaning 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 PHYEND and DISABLE. As illustrated in the following figure, RADIO will by default transmit an unmodulated carrier between READY and START, and between PHYEND and DISABLED.

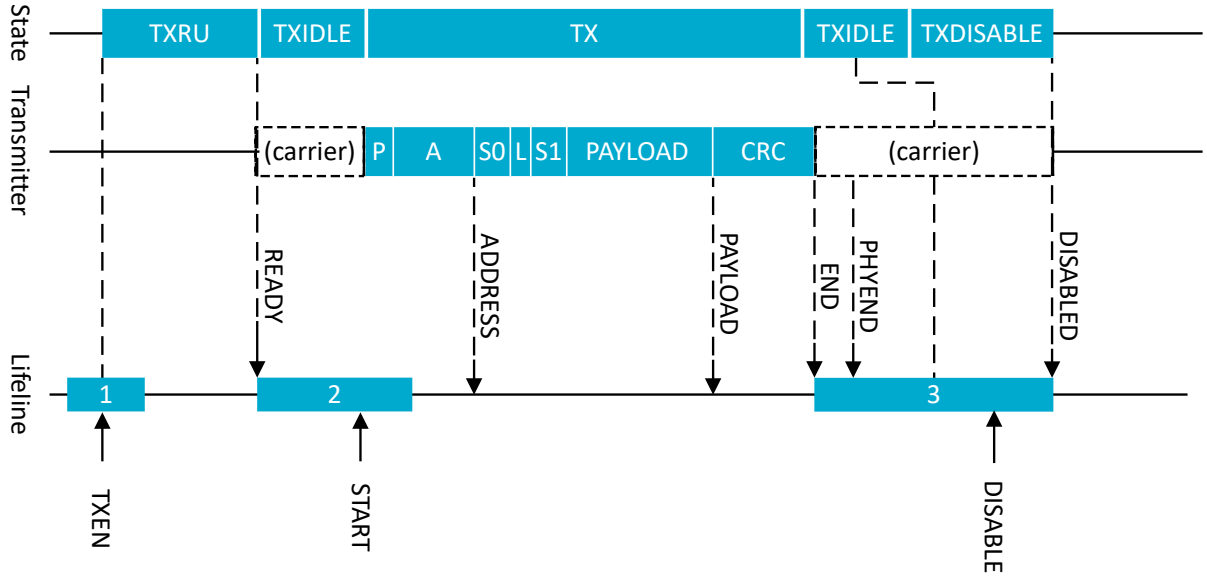


Figure 72: Transmit sequence

The following figure shows a transmit sequence where no delay is introduced. RADIO is configured to use shortcuts between READY and START, and between PHYEND and DISABLE.

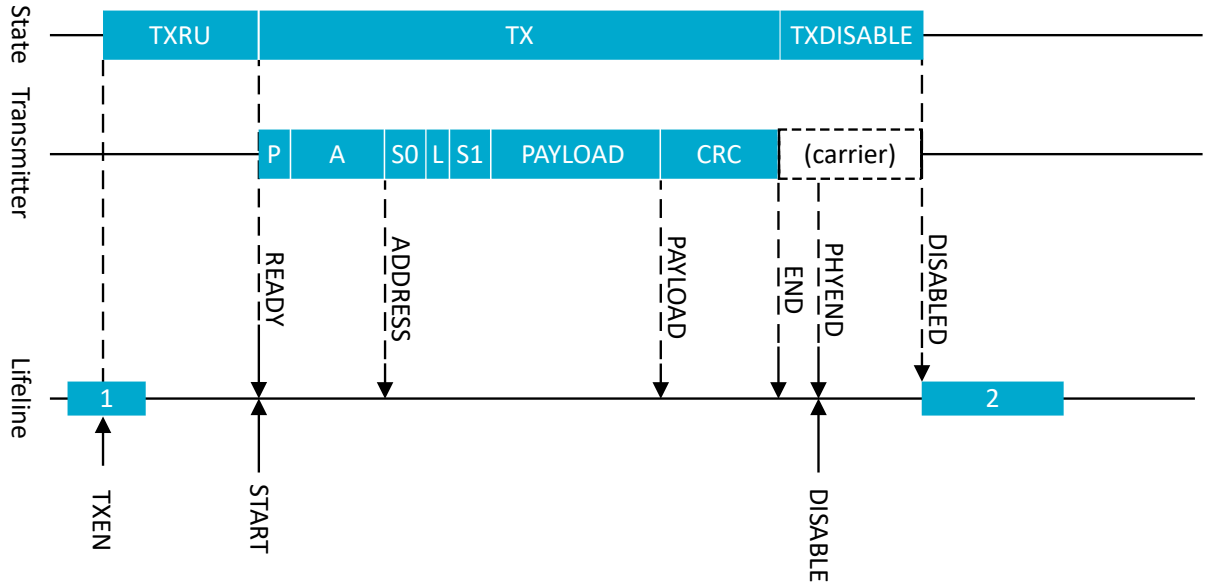


Figure 73: Transmit sequence using shortcuts to avoid delays

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.

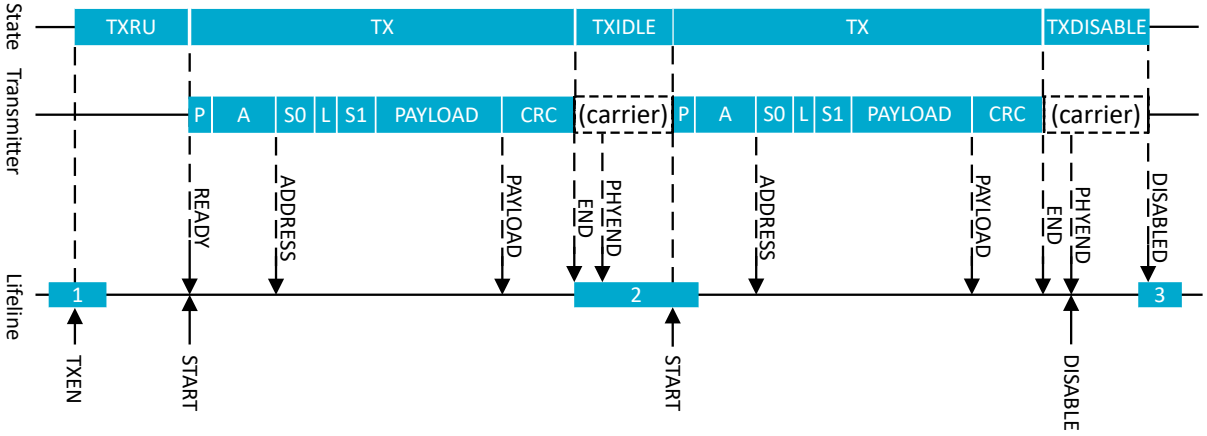


Figure 74: Transmission of multiple packets

8.12.7 Receive sequence

Before RADIO is able to receive a packet, it must first ramp up in RX mode. See RXRU in Radio states without PLL on page 282 and Receive sequence on page 285 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 without PLL on page 282, the START task can first be triggered after RADIO has entered the RXIDLE state.

The following figure shows a single packet reception where the CPU manually triggers the tasks needed to control the flow of RADIO, meaning 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 PHYEND 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.

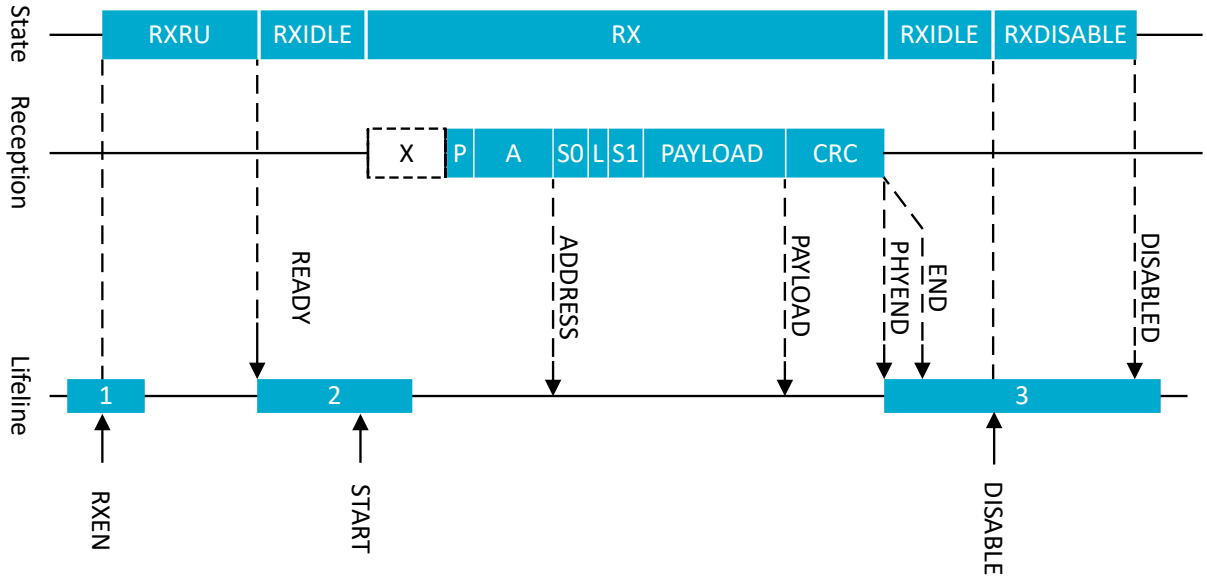


Figure 75: Receive sequence

The following figure shows a receive sequence where no delay is introduced. RADIO is configured to use shortcuts between READY and START, and between PHYEND and DISABLE.

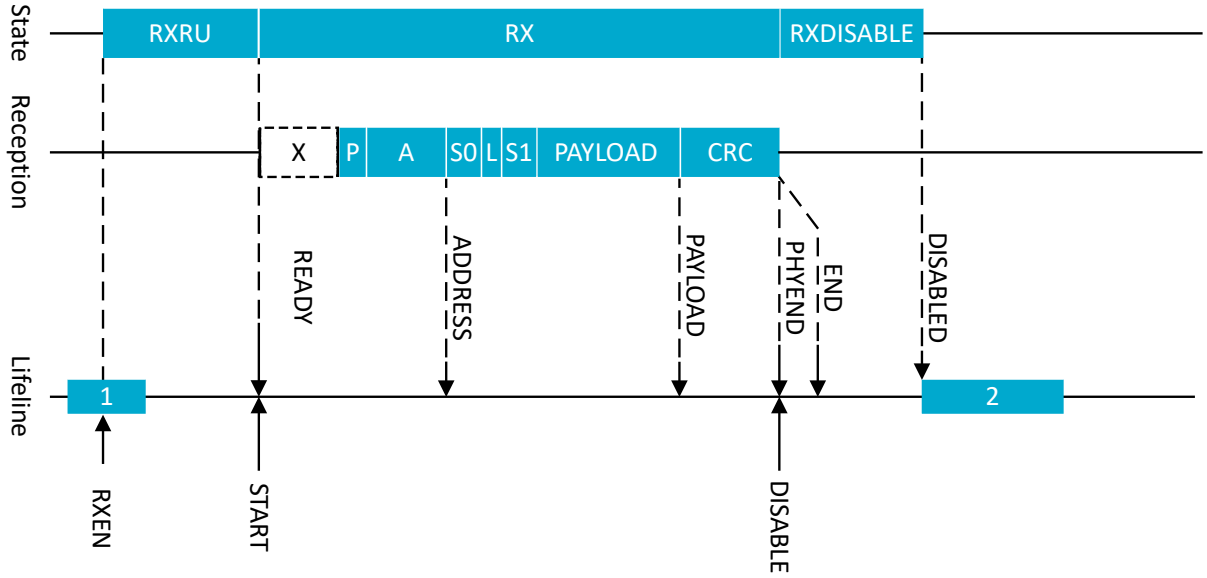


Figure 76: Receive sequence using shortcuts to avoid delays

RADIO can receive consecutive packets without having to disable and re-enable RADIO between packets, as illustrated in the following figure.

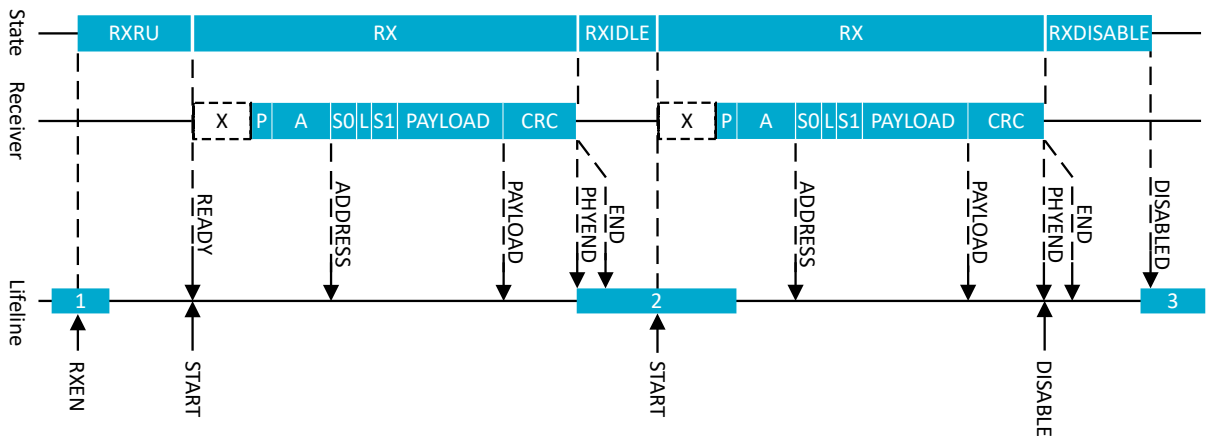


Figure 77: Reception of multiple packets

8.12.8 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 continuously measured and the value filtered using a single-pole IIR filter. After a signal level change, the RSSI will settle after approximately $RSSI_{SETTLE}$.

Sampling 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 by triggering the RXEN task and the reception has to be started with the READY event followed by START task.

8.12.9 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 the RADIO peripheral's turnaround time (meaning the time needed to switch off the receiver and then switch on the transmitter). The TIFS register can be written any time before the last bit on air is received.

This timing is illustrated in the following figure.

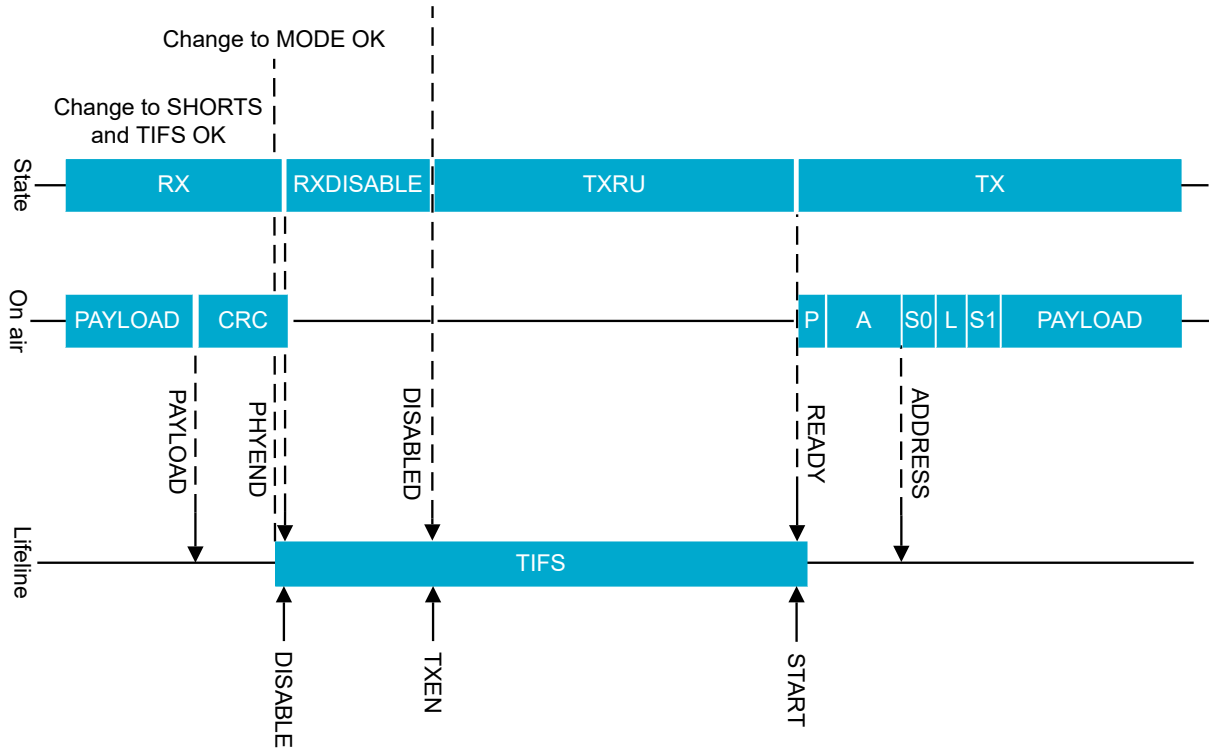


Figure 78: IFS timing detail

The TIFS duration starts after the last bit on air (at the PHYEND event), and elapses with the first bit being transmitted on air (just after READY event).

TIFS is only enforced if the shortcuts PHYEND to DISABLE and DISABLED to TXEN or PHYEND to DISABLE and DISABLED to RXEN are enabled. The short READY to START must also be enabled. In these configurations, TXEN or RXEN is automatically delayed to achieve the configured interframe spacing.

TIFS is qualified for use in Bluetooth Low Energy 1 Mbps and 2 Mbps modes, using the Legacy 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.

8.12.10 Device address match

The device address match feature is tailored for device filtering in Bluetooth Low Energy and similar implementations.

This feature enables on-the-fly device address matching while receiving a packet over-the-air. This feature only works in RX 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 the device filtering procedure.

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.

8.12.11 Bit counter

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

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

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 bits specified in register BCC. The bit counter will continue to count bits until the DISABLED event is generated or until the BCSTOP task is triggered. After the BCMATCH event, the CPU can reconfigure the value BCC 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 for the tasks BCSTOP, STOP, or DISABLE, or the event END.

The following figure shows how the bit counter can be used to generate the 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.

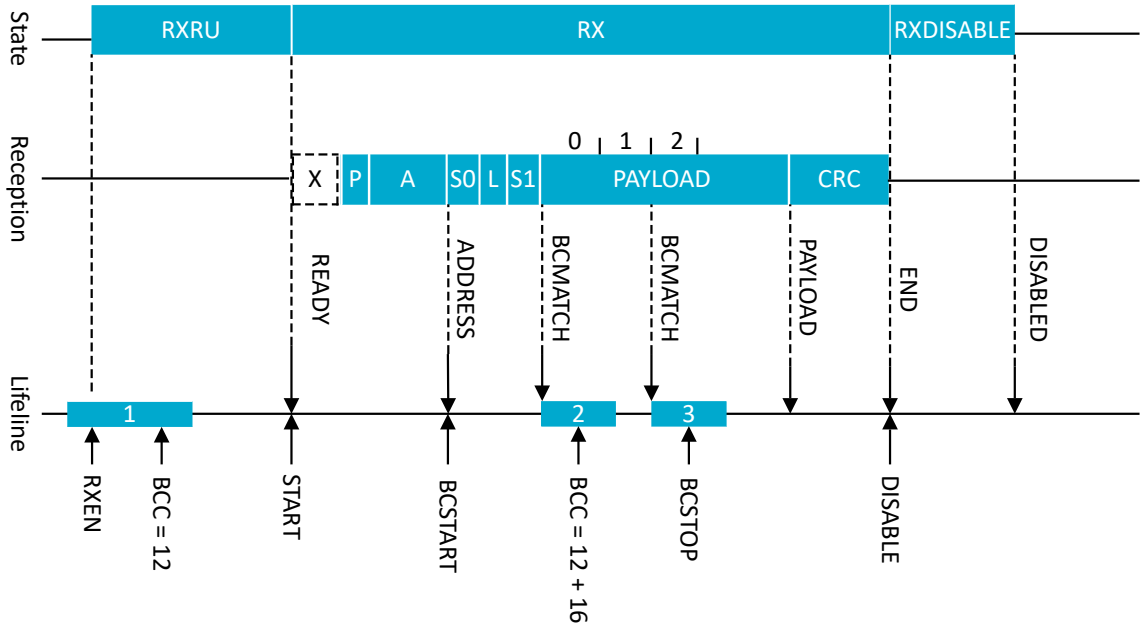


Figure 79: Bit counter example

RXRU assumes that the total combined length of S0, L, and S1 is 12 bits.

8.12.12 EasyDMA

RADIO uses EasyDMA to read and write packets to RAM without CPU involvement.

As illustrated in RADIO block diagram on page 278, the RADIO peripheral'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 the DISABLE task is done.

The structure of a packet is described in detail in Packet configuration on page 278. 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 listed fields in the frame is configurable (see Packet configuration on page 278), 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:

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

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 exceeds 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 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 RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 11 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 the DISABLE task is done.

8.12.13 Registers

Instances

Instance	Domain	Base address	Description
RADIO	GLOBAL	0x4008A000	2.4 GHz radio RADIO

Configuration

Instance	Domain	Configuration
RADIO	GLOBAL	For the PSEL registers, use only dedicated pins on port P1
		No IEEE 802.15.4 support
		No long range (LE Coded PHY) support
		No direction finding support (DFE)
		No channel sounding support (CSTONES)
		No internal instantiation of DmaChannelPeripheral

Register overview

Register	Offset	Description
TASKS_TXEN	0x000	Enable RADIO in TX mode
TASKS_RXEN	0x004	Enable RADIO in RX mode
TASKS_START	0x008	Start RADIO
TASKS_STOP	0x00C	Stop RADIO
TASKS_DISABLE	0x010	Disable RADIO
TASKS_RSSISTART	0x014	Start the RSSI and take one single sample of the receive signal strength
TASKS_BCSTART	0x018	Start the bit counter
TASKS_BCSTOP	0x01C	Stop the bit counter
TASKS_PLLEN	0x06C	Enable RADIO in PLL mode (standby for either TX or RX)
TASKS_SOFTRESET	0x0A4	Reset all public registers, but with these exceptions: DMA registers and EVENT/INTEN/SUBSCRIBE/PUBLISH registers. Only to be used in DISABLED state.
SUBSCRIBE_TXEN	0x100	Subscribe configuration for task TXEN
SUBSCRIBE_RXEN	0x104	Subscribe configuration for task RXEN
SUBSCRIBE_START	0x108	Subscribe configuration for task START
SUBSCRIBE_STOP	0x10C	Subscribe configuration for task STOP
SUBSCRIBE_DISABLE	0x110	Subscribe configuration for task DISABLE
SUBSCRIBE_RSSISTART	0x114	Subscribe configuration for task RSSISTART
SUBSCRIBE_BCSTART	0x118	Subscribe configuration for task BCSTART
SUBSCRIBE_BCSTOP	0x11C	Subscribe configuration for task BCSTOP
SUBSCRIBE_PLLEN	0x16C	Subscribe configuration for task PLLEN
SUBSCRIBE_SOFTRESET	0x1A4	Subscribe configuration for task SOFTRESET
EVENTS_READY	0x200	RADIO has ramped up and is ready to be started
EVENTS_TXREADY	0x204	RADIO has ramped up and is ready to be started TX path
EVENTS_RXREADY	0x208	RADIO has ramped up and is ready to be started RX path
EVENTS_ADDRESS	0x20C	Address sent or received
EVENTS_PAYLOAD	0x214	Packet payload sent or received
EVENTS_END	0x218	Memory access for packet data has been completed
EVENTS_PHYEND	0x21C	The last bit is sent on air or last bit is received
EVENTS_DISABLED	0x220	RADIO has been disabled
EVENTS_DEVMATCH	0x224	A device address match occurred on the last received packet
EVENTS_DEVMISS	0x228	No device address match occurred on the last received packet
EVENTS_CRCOK	0x22C	Packet received with CRC ok
EVENTS_CRCERROR	0x230	Packet received with CRC error
EVENTS_BCMATCH	0x238	Bit counter reached bit count value
EVENTS_SYNC	0x258	Initial sync detected
EVENTS_PLLREADY	0x2B0	PLL has settled and RADIO is ready to be enabled in either TX or RX mode
EVENTS_RXADDRESS	0x2BC	Address received
PUBLISH_READY	0x300	Publish configuration for event READY

Register	Offset	Description
PUBLISH_TXREADY	0x304	Publish configuration for event TXREADY
PUBLISH_RXREADY	0x308	Publish configuration for event RXREADY
PUBLISH_ADDRESS	0x30C	Publish configuration for event ADDRESS
PUBLISH_PAYLOAD	0x314	Publish configuration for event PAYLOAD
PUBLISH_END	0x318	Publish configuration for event END
PUBLISH_PHYEND	0x31C	Publish configuration for event PHYEND
PUBLISH_DISABLED	0x320	Publish configuration for event DISABLED
PUBLISH_DEVMATCH	0x324	Publish configuration for event DEVMATCH
PUBLISH_DEVMISS	0x328	Publish configuration for event DEVMISS
PUBLISH_CRCOK	0x32C	Publish configuration for event CRCOK
PUBLISH_CRCERROR	0x330	Publish configuration for event CRCERROR
PUBLISH_BCMATCH	0x338	Publish configuration for event BCMATCH
PUBLISH_SYNC	0x358	Publish configuration for event SYNC
PUBLISH_PLLREADY	0x3B0	Publish configuration for event PLLREADY
PUBLISH_RXADDRESS	0x3BC	Publish configuration for event RXADDRESS
SHORTS	0x400	Shortcuts between local events and tasks
INTENSET00	0x488	Enable interrupt
INTENSET01	0x48C	Enable interrupt
INTENCLR00	0x490	Disable interrupt
INTENCLR01	0x494	Disable interrupt
INTENSET10	0x4A8	Enable interrupt
INTENSET11	0x4AC	Enable interrupt
INTENCLR10	0x4B0	Disable interrupt
INTENCLR11	0x4B4	Disable interrupt
MODE	0x500	Data rate and modulation
PHYENDTXDELAY	0x518	Configurable delay of PHYEND event for TX
STATE	0x520	Current radio state
DATAWHITE	0x540	Data whitening configuration
TIMING	0x704	Timing
FREQUENCY	0x708	Frequency
TXPOWER	0x710	Output power
TIFS	0x714	Interframe spacing in µs
RSSISAMPLE	0x718	RSSI sample
RXGAIN.CONFIG	0x7D4	Override configuration of receiver gain control loop
FREQFINETUNE	0x0804	Fine tuning of the RF frequency
FECONFIG	0x908	Config register
CFO_STAT	0xB00	Carrier freq. offset estimate
DBCCORR	0xB40	Correlator thresholds
CRCSTATUS	0xE0C	CRC status
RXMATCH	0xE10	Received address
RXCRC	0xE14	CRC field of previously received packet
DAI	0xE18	Device address match index
PDUSTAT	0xE1C	Payload status
PCNF0	0xE20	Packet configuration register 0
PCNF1	0xE28	Packet configuration register 1
BASE0	0xE2C	Base address 0
BASE1	0xE30	Base address 1
PREFIX0	0xE34	Prefixes bytes for logical addresses 0-3
PREFIX1	0xE38	Prefixes bytes for logical addresses 4-7
TXADDRESS	0xE3C	Transmit address select
RXADDRESSES	0xE40	Receive address select
CRCCNF	0xE44	CRC configuration
CRCPOLY	0xE48	CRC polynomial

Register	Offset	Description
CRCINIT	0xE4C	CRC initial value
DAB[n]	0xE50	Device address base segment n
DAP[n]	0xE70	Device address prefix n
DACNF	0xE90	Device address match configuration
BCC	0xE94	Bit counter compare
MHRMATCHCONF	0xEB4	Search pattern configuration
MHRMATCHMASK	0xEB8	Pattern mask
PACKETPTR	0xED0	Packet pointer

8.12.13.1 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

8.12.13.2 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

8.12.13.3 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

8.12.13.4 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

8.12.13.5 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

8.12.13.6 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

8.12.13.7 TASKS_BCSTART

Address offset: 0x018

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

8.12.13.8 TASKS_BCSTOP

Address offset: 0x01C

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

8.12.13.9 TASKS_PLLEN

Address offset: 0x06C

Enable RADIO in PLL mode (standby for either TX or RX)

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_PLLEN			Enable RADIO in PLL mode (standby for either TX or RX)
			Trigger	1	Trigger task

8.12.13.10 TASKS_SOFTRESET

Address offset: 0x0A4

Reset all public registers, but with these exceptions: DMA registers and EVENT/INTEN/SUBSCRIBE/PUBLISH registers. Only to be used in DISABLED 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_SOFTRESET			Reset all public registers, but with these exceptions: DMA registers and EVENT/INTEN/SUBSCRIBE/PUBLISH registers. Only to be used in DISABLED state.
			Trigger	1	Trigger task

8.12.13.11 SUBSCRIBE_TXEN

Address offset: 0x100

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		[0..255]	DPPI channel that task TXEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.12 SUBSCRIBE_RXEN

Address offset: 0x104

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		[0..255]	DPPI channel that task RXEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.13 SUBSCRIBE_START

Address offset: 0x108

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.14 SUBSCRIBE_STOP

Address offset: 0x10C

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.15 SUBSCRIBE_DISABLE

Address offset: 0x110

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task DISABLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.16 SUBSCRIBE_RSSISTART

Address offset: 0x114

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RSSISTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.17 SUBSCRIBE_BCSTART

Address offset: 0x118

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task BCSTART will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.18 SUBSCRIBE_BCSTOP

Address offset: 0x11C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task BCSTOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.19 SUBSCRIBE_PLLEN

Address offset: 0x16C

Subscribe configuration for task PLLEN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task PLLEN will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.20 SUBSCRIBE_SOFTRESET

Address offset: 0x1A4

Subscribe configuration for task SOFTRESET

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task SOFTRESET will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.12.13.21 EVENTS_READY

Address offset: 0x200

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

8.12.13.22 EVENTS_TXREADY

Address offset: 0x204

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

8.12.13.23 EVENTS_RXREADY

Address offset: 0x208

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

8.12.13.24 EVENTS_ADDRESS

Address offset: 0x20C

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

8.12.13.25 EVENTS_PAYLOAD

Address offset: 0x214

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

8.12.13.26 EVENTS_END

Address offset: 0x218

Memory access for packet data has been completed

In TX: Last byte to be transmitted has been fetched from RAM

In RX: Last byte received on air has been stored 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_END			Memory access for packet data has been completed In TX: Last byte to be transmitted has been fetched from RAM In RX: Last byte received on air has been stored to RAM
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.12.13.27 EVENTS_PHYEND

Address offset: 0x21C

The last bit is sent on air or last bit is 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_PHYEND			The last bit is sent on air or last bit is received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.12.13.28 EVENTS_DISABLED

Address offset: 0x220

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

8.12.13.29 EVENTS_DEVMATCH

Address offset: 0x224

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

8.12.13.30 EVENTS_DEVMISS

Address offset: 0x228

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

8.12.13.31 EVENTS_CRCOK

Address offset: 0x22C

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

8.12.13.32 EVENTS_CRCERROR

Address offset: 0x230

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

8.12.13.33 EVENTS_BCMATCH

Address offset: 0x238

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

8.12.13.34 EVENTS_SYNC

Address offset: 0x258

Initial sync detected

For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode.

It is also possible that the event is not generated, or not generated before the ADDRESS 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_SYNC			Initial sync detected
					For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode.
					It is also possible that the event is not generated, or not generated before the ADDRESS event.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.12.13.35 EVENTS_PLLREADY

Address offset: 0x2B0

PLL has settled and RADIO is ready to be enabled in either TX or 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	RW	EVENTS_PLLREADY			PLL has settled and RADIO is ready to be enabled in either TX or RX mode
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.12.13.36 EVENTS_RXADDRESS

Address offset: 0x2BC

Address 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	EVENTS_RXADDRESS			Address received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.12.13.37 PUBLISH_READY

Address offset: 0x300

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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.38 PUBLISH_TXREADY

Address offset: 0x304

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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event TXREADY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.39 PUBLISH_RXREADY

Address offset: 0x308

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**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event RXREADY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.40 PUBLISH_ADDRESS

Address offset: 0x30C

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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event ADDRESS will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.41 PUBLISH_PAYLOAD

Address offset: 0x314

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event PAYLOAD will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.42 PUBLISH_END

Address offset: 0x318

Publish configuration for event END

In TX: Last byte to be transmitted has been fetched from RAM

In RX: Last byte received on air has been stored 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				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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.43 PUBLISH_PHYEND

Address offset: 0x31C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event PHYEND will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.44 PUBLISH_DISABLED

Address offset: 0x320

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DISABLED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.45 PUBLISH_DEVMATCH

Address offset: 0x324

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DEVMATCH will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.46 PUBLISH_DEVMISS

Address offset: 0x328

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DEVMISS will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.47 PUBLISH_CRCOK

Address offset: 0x32C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CRCOK will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.48 PUBLISH_CRCERROR

Address offset: 0x330

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CRCERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.49 PUBLISH_BCMATCH

Address offset: 0x338

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BCMATCH will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.50 PUBLISH_SYNC

Address offset: 0x358

Publish configuration for event SYNC

For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode.

It is also possible that the event is not generated, or not generated before the ADDRESS 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				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		[0..255]	DPPI channel that event SYNC will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.51 PUBLISH_PLLREADY

Address offset: 0x3B0

Publish configuration for event PLLREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event PLLREADY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.52 PUBLISH_RXADDRESS

Address offset: 0x3BC

Publish configuration for event RXADDRESS

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event RXADDRESS will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.12.13.53 SHORTS

Address offset: 0x400

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				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	READY_START			Shortcut between event READY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	DISABLED_TXEN			Shortcut between event DISABLED and task TXEN
			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				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	DISABLED_RXEN			Shortcut between event DISABLED and task RXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ADDRESS_RSSISTART			Shortcut between event ADDRESS and task RSSISTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	END_START			Shortcut between event END and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	ADDRESS_BCSTART			Shortcut between event ADDRESS and task BCSTART
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G	RW	PHYEND_PLLEN			Shortcut between event PHYEND and task PLLEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
H	RW	TXREADY_START			Shortcut between event TXREADY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
I	RW	RXREADY_START			Shortcut between event RXREADY and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
J	RW	PHYEND_DISABLE			Shortcut between event PHYEND and task DISABLE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
K	RW	PHYEND_START			Shortcut between event PHYEND and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
L	RW	PLLREADY_TXEN			Shortcut between event PLLREADY and task TXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
M	RW	PLLREADY_RXEN			Shortcut between event PLLREADY and task RXEN
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.12.13.54 INTENSET00

Address offset: 0x488

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				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 W1S	READY			Write '1' to enable interrupt for event READY
			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										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
B	RW W1S	TXREADY			Write '1' to enable interrupt for event TXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	RXREADY			Write '1' to enable interrupt for event RXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	ADDRESS			Write '1' to enable interrupt for event ADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	PAYLOAD			Write '1' to enable interrupt for event PAYLOAD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	END			Write '1' to enable interrupt for event END In TX: Last byte to be transmitted has been fetched from RAM In RX: Last byte received on air has been stored to RAM
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	PHYEND			Write '1' to enable interrupt for event PHYEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1S	DISABLED			Write '1' to enable interrupt for event DISABLED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1S	DEVMATCH			Write '1' to enable interrupt for event DEVMATCH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1S	DEVMISS			Write '1' to enable interrupt for event DEVMISS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1S	CRCOK			Write '1' to enable interrupt for event CRCOK
			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				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
L	RW W1S	CRCERROR			Write '1' to enable interrupt for event CRCERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	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
N	RW W1S	SYNC			Write '1' to enable interrupt for event SYNC For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode. It is also possible that the event is not generated, or not generated before the ADDRESS event.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.55 INTENSET01

Address offset: 0x48C

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 W1S	PLLREADY			Write '1' to enable interrupt for event PLLREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	RXADDRESS			Write '1' to enable interrupt for event RXADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.56 INTENCLR00

Address offset: 0x490

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										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 W1C	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	TXREADY			Write '1' to disable interrupt for event TXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	RXREADY			Write '1' to disable interrupt for event RXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	ADDRESS			Write '1' to disable interrupt for event ADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	PAYLOAD			Write '1' to disable interrupt for event PAYLOAD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	END			Write '1' to disable interrupt for event END In TX: Last byte to be transmitted has been fetched from RAM In RX: Last byte received on air has been stored to RAM
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1C	PHYEND			Write '1' to disable interrupt for event PHYEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1C	DISABLED			Write '1' to disable interrupt for event DISABLED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1C	DEVMATCH			Write '1' to disable interrupt for event DEVMATCH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1C	DEVMISS			Write '1' to disable interrupt for event DEVMISS
			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				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
K	RW W1C	CRCOK			Write '1' to disable interrupt for event CRCOK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	CRCERROR			Write '1' to disable interrupt for event CRCERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	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
N	RW W1C	SYNC			Write '1' to disable interrupt for event SYNC For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode. It is also possible that the event is not generated, or not generated before the ADDRESS event.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.57 INTENCLR01

Address offset: 0x494

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 W1C	PLLREADY			Write '1' to disable interrupt for event PLLREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	RXADDRESS			Write '1' to disable interrupt for event RXADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.58 INTENSET10

Address offset: 0x4A8

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										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 W1S	READY			Write '1' to enable interrupt for event READY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	TXREADY			Write '1' to enable interrupt for event TXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	RXREADY			Write '1' to enable interrupt for event RXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	ADDRESS			Write '1' to enable interrupt for event ADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	PAYLOAD			Write '1' to enable interrupt for event PAYLOAD
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	END			Write '1' to enable interrupt for event END In TX: Last byte to be transmitted has been fetched from RAM In RX: Last byte received on air has been stored to RAM
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	PHYEND			Write '1' to enable interrupt for event PHYEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1S	DISABLED			Write '1' to enable interrupt for event DISABLED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1S	DEVMATCH			Write '1' to enable interrupt for event DEVMATCH
			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										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
J	RW W1S	DEVMISS			Write '1' to enable interrupt for event DEVMISS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1S	CRCOK			Write '1' to enable interrupt for event CRCOK
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1S	CRCERROR			Write '1' to enable interrupt for event CRCERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	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
N	RW W1S	SYNC			Write '1' to enable interrupt for event SYNC For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode. It is also possible that the event is not generated, or not generated before the ADDRESS event.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.59 INTENSET11

Address offset: 0x4AC

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 W1S	PLLREADY			Write '1' to enable interrupt for event PLLREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	RXADDRESS			Write '1' to enable interrupt for event RXADDRESS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.60 INTENCLR10

Address offset: 0x4B0

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				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 W1C	READY			Write '1' to disable interrupt for event READY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	TXREADY			Write '1' to disable interrupt for event TXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	RXREADY			Write '1' to disable interrupt for event RXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	ADDRESS			Write '1' to disable interrupt for event ADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	PAYLOAD			Write '1' to disable interrupt for event PAYLOAD
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	END			Write '1' to disable interrupt for event END In TX: Last byte to be transmitted has been fetched from RAM In RX: Last byte received on air has been stored to RAM
			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	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
G	RW W1C	PHYEND			Write '1' to disable interrupt for event PHYEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1C	DISABLED			Write '1' to disable interrupt for event DISABLED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1C	DEVMATCH			Write '1' to disable interrupt for event DEVMATCH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1C	DEVMISS			Write '1' to disable interrupt for event DEVMISS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1C	CRCOK			Write '1' to disable interrupt for event CRCOK
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	CRCERROR			Write '1' to disable interrupt for event CRCERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	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
N	RW W1C	SYNC			Write '1' to disable interrupt for event SYNC For MODE=Nrf_1Mbit, Nrf_2Mbit, Ble_1Mbit, or Ble_2Mbit: A possible preamble and the first two bytes of the address field has been received. The event can be generated falsely also in this mode. It is also possible that the event is not generated, or not generated before the ADDRESS event.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.61 INTENCLR11

Address offset: 0x4B4

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 W1C	PLLREADY			Write '1' to disable interrupt for event PLLREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	RXADDRESS			Write '1' to disable interrupt for event RXADDRESS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.12.13.62 MODE

Address offset: 0x500

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
			Nrf_4Mbit_0BT6	9	4 Mbps Nordic proprietary radio mode (BT=0.6/h=0.5)
			Nrf_4Mbit_0BT4	10	4 Mbps Nordic proprietary radio mode (BT=0.4/h=0.5)

8.12.13.63 PHYENDTXDELAY

Address offset: 0x518

Configurable delay of PHYEND event for TX

There are separate values for each on-air bit rate. The maximum supported value is 3.5 us (unit is 0.5 us)

This register will not be reset by the SOFTRESET 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				D D D C C C B B B A A A
Reset 0x00000421				**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 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	RATE4M			For modes with 4 Mbps on-air bit rate, unit is 2 bit periods (Nrf_4Mbit0_5 and Nrf_4Mbit0_25 modes) Default 1 (0.5 us = 2 on-air bit periods)
B	RW	RATE2M			For modes with 2 Mbps on-air bit rate, unit is 1 bit period (Nrf_2Mbit, Ble_2Mbit, and Ieee802154_250Kbit modes) Default 2 (1 us = 2 on-air bit periods)
C	RW	RATE1M			For modes with 1 Mbps on-air bit rate, unit is 1/2 bit period (Nrf_1Mbit and Ble_1Mbit modes) Default 4 (2 us = 2 on-air bit periods)
D	RW	RATE250K			For modes with 250 kbps on-air bit rate, unit is 1/8 bit period (Nrf_250Kbit mode) Default 0

8.12.13.64 STATE

Address offset: 0x520

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 RXDISABLE state
			Settle	5	RADIO is in the SETTLE state
			Pll	6	RADIO is in the PLL 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 TXDISABLE state

8.12.13.65 DATAWHITE

Address offset: 0x540

Data whitening configuration

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B B B A A A A A A A A
Reset 0x00890040				**0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 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	IV			Whitening initial value Data whitening initial value.
B	RW	POLY			Whitening polynomial Data whitening polynomial. Bit 0 is always interpreted as 1.

8.12.13.66 TIMING

Address offset: 0x704

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				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	RU			Ramp-up time
			Legacy	0	Legacy ramp-up time
			Fast	1	Fast ramp-up (default)

8.12.13.67 FREQUENCY

Address offset: 0x708

Frequency

Note: This register is double-buffered, and will only take effect on the next TXEN/RXEN/PLLEN task trigger

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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			Radio channel frequency. Frequency = 2400 + FREQUENCY (MHz).
B	RW	MAP			Channel map selection. 0: Channel map between 2400 MHZ to 2500 MHz, Frequency = 2400 + FREQUENCY (MHz). 1: Channel map between 2360 MHZ to 2460 MHz, Frequency = 2360 + FREQUENCY (MHz).

8.12.13.68 TXPOWER

Address offset: 0x710

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 A A
Reset 0x00000013	**0 0 0 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**

ID	R/W	Field	Value ID	Value	Description
A	RW	TXPOWER			RADIO output power Output power, see value mapping to dBm below. **Note:** This register is double-buffered, and will only take effect on the next TXEN/RXEN/PLLEN task trigger
			MaxdBm	0x3F	+8 dBm
			Pos8dBm	0x3F	+8 dBm
			Pos7dBm	0x39	+7 dBm
			Pos6dBm	0x33	+6 dBm
			Pos5dBm	0x2D	+5 dBm
			Pos4dBm	0x28	+4 dBm
			Pos3dBm	0x23	+3 dBm
			Pos2dBm	0x1F	+2 dBm
			Pos1dBm	0x1B	+1 dBm
			0dBm	0x18	0 dBm
			Neg1dBm	0x15	-1 dBm
			Neg2dBm	0x13	-2 dBm
			Neg3dBm	0x11	-3 dBm
			Neg4dBm	0xF	-4 dBm
			Neg5dBm	0xD	-5 dBm
			Neg6dBm	0xB	-6 dBm
			Neg7dBm	0xA	-7 dBm
			Neg8dBm	0x9	-8 dBm
			Neg9dBm	0x8	-9 dBm
			Neg10dBm	0x7	-10 dBm
			Neg12dBm	0x6	-12 dBm
			Neg14dBm	0x5	-14 dBm
			Neg16dBm	0x4	-16 dBm
			Neg18dBm	0x3	-18 dBm
			Neg20dBm	0x2	-20 dBm
			Neg22dBm	0x2	-22 dBm
			Neg28dBm	0x1	-28 dBm
			Neg40dBm	0x130	-40 dBm
			Neg46dBm	0x110	-46 dBm
			MindBm	0x0	Minimum output power

8.12.13.69 TIFS

Address offset: 0x714

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 A A
Reset 0x00000000	**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 us. 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.

8.12.13.70 RSSISAMPLE

Address offset: 0x718

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 0x0000007F				**0**	**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**	**1**
ID	R/W	Field	Value ID	Value																															
A	R	RSSISAMPLE																											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.											

8.12.13.71 RXGAIN.CONFIG

Address offset: 0x7D4

Override configuration of receiver gain control loop

Overriding the default values can result in unpredictable behavior

This register will not be reset by the SOFTRESET 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							D					C	C	B	A	A	A	A																	
Reset 0x801230C3				**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**
ID	R/W	Field	Value ID	Value																															
A	RW	AGCAAFOVERRIDE																											Override value for AAF						
B	RW	AGCMIXOVERRIDE																											Override value for MIX						
C	RW	AGCLNAOVERRIDE																											Override value for LNA						
D	RW	AGCOVERRIDEGAIN																											Enable AGC override						
			NoOverride	0																									AGC takes control over all gains						
			Override	1																									Manual control of AAF, MIX, and LNA gain settings						

8.12.13.72 FREQFINETUNE

Address offset: 0x0804

Fine tuning of the RF frequency

Receiver sensitivity may be degraded when operating on 2414, 2415, 2430, 2431, 2446, 2447, 2462, 2463, 2478 or 2479 MHz with a small but non-zero FREQFINETUNE value

This register will not be reset by the SOFTRESET 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	FREQFINETUNE																											Twos-complement number for fine-tuning the frequency. The step size is 488.28125 Hz, giving a range from -1 MHz to (one step short of) +1 MHz.											

8.12.13.73 FECONFIG

Address offset: 0x908

Config register

This register will not be reset by the SOFTRESET 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 0x10800005				**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**	**1**	**0**	**1**
ID	R/W	Field	Value ID	Value																															
A	RW	SCALERMODE																																	
			Disabled	0																															
			Enabled	1																															

8.12.13.74 CFO_STAT

Address offset: 0xB00

Carrier freq. offset estimate

This register will not be reset by the SOFTRESET 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																															
A	R	SYNCOK																																	
			SyncNotOK	0																															
			SyncOk	1																															

8.12.13.75 DBCCORR

Address offset: 0xB40

Correlator thresholds

This register will not be reset by the SOFTRESET 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
Reset 0x1FFFFF90				**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
ID	R/W	Field	Value ID	Value																															
A	RW	TH																																	

8.12.13.76 CRCSTATUS

Address offset: 0xE0C

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																															
A	R	CRCSTATUS																																	
			CRCError	0																															
			CRCOk	1																															

8.12.13.77 RXMATCH

Address offset: 0xE10

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

8.12.13.78 RXCRC

Address offset: 0xE14

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

8.12.13.79 DAI

Address offset: 0xE18

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

8.12.13.80 PDUSTAT

Address offset: 0xE1C

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				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

8.12.13.81 PCNF0

Address offset: 0xE20

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				F E E D D C C C C 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	LFLEN			Length on air of LENGTH field in number of bits.
B	RW	S0LEN			Length on air of S0 field in number of bytes.
C	RW	S1LEN			Length on air of S1 field in number of bits.
D	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
E	RW	PLEN			Length of preamble on air. Decision point: TASKS_START task
			8bit	0	8-bit preamble
			16bit	1	16-bit preamble
F	RW	CRCINC			Indicates if LENGTH field contains CRC or not
			Exclude	0	LENGTH does not contain CRC
			Include	1	LENGTH includes CRC

8.12.13.82 PCNF1

Address offset: 0xE28

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				F 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
F	RW	WHITEOFFSET			If whitening is enabled S0 can be configured to be excluded from whitening
			Include	0	S0 included in whitening

Bit number	31	30	29	28	27	26	25	24	23	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	C	C	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
			Exclude	1	S0 excluded from whitening

8.12.13.83 BASE0

Address offset: 0xE2C

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

8.12.13.84 BASE1

Address offset: 0xE30

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

8.12.13.85 PREFIX0

Address offset: 0xE34

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

8.12.13.86 PREFIX1

Address offset: 0xE38

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

8.12.13.87 TXADDRESS

Address offset: 0xE3C

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

8.12.13.88 RXADDRESSES

Address offset: 0xE40

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

8.12.13.89 CRCCNF

Address offset: 0xE44

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 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			CRC length in number of bytes.
			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			Control whether CRC calculation skips the address field. Other fields can also be skipped.
			Include	0	CRC calculation includes address field
			Skip	1	CRC calculation starting at first byte after address field.
			Ieee802154	2	CRC calculation starting at first byte after length field (as per 802.15.4 standard).
			SkipS0	3	CRC calculation starting at first byte after S0 field.
			SkipS1	4	CRC calculation starting at first byte after S1 field.

8.12.13.90 CRCPOLY

Address offset: 0xE48

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
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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 .

8.12.13.91 CRCINIT

Address offset: 0xE4C

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
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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

8.12.13.92 DAB[n] (n=0..7)

Address offset: 0xE50 + (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

8.12.13.93 DAP[n] (n=0..7)

Address offset: 0xE70 + (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

8.12.13.94 DACNF

Address offset: 0xE90

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																															
A-H	RW	ENA[i] (i=0..7)																																	
			Disabled	0																															
			Enabled	1																															
I-P	RW	TXADD[i] (i=0..7)																																	

ID	R/W	Field	Description
A-H	RW	ENA[i] (i=0..7)	Enable or disable device address matching using device address i
		Disabled	Disabled
		Enabled	Enabled
I-P	RW	TXADD[i] (i=0..7)	TxAdd for device address i

8.12.13.95 BCC

Address offset: 0xE94

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																															
A	RW	BCC																																	

ID	R/W	Field	Description
A	RW	BCC	Bit counter compare Bit counter compare register

8.12.13.96 MHRMATCHCONF

Address offset: 0xEB4

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																															
A	RW	MHRMATCHCONF																																	

ID	R/W	Field	Description
A	RW	MHRMATCHCONF	Search pattern configuration

8.12.13.97 MHRMATCHMASK

Address offset: 0xEB8

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																															
A	RW	MHRMATCHMASK																																	

ID	R/W	Field	Description
A	RW	MHRMATCHMASK	Pattern mask

8.12.13.98 PACKETPTR

Address offset: 0xED0

Packet pointer

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

ID	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.

8.13 SAADC — Successive approximation analog-to-digital converter

The SAADC peripheral is a differential successive approximation register (SAR) analog-to-digital converter.

The main features of SAADC are the following:

- Operation modes
  - 10-bit arithmetic resolution in single-ended mode, 11-bit arithmetic resolution in differential mode
  - Multiple 20 kHz inputs with maximum sample rate of 1 Msps (all inputs)
  - One fast input up to 500 kHz (Nyquist) frequency with maximum sample rate 1 Msps (only with AIN0 positive and AIN1 negative)
  - Improved accuracy with oversampling
  - Configurable 8/10/12/14 bit output resolution
- Multiple analog inputs:
  - GPIO pins with analog function (input range 0 to VDD)
  - VDD (divided down to a valid range)
- Up to four fixed gain input channels with 20 kHz bandwidth:
  - One input per single-ended channel, and two inputs per differential channel
  - Scan mode configurable with both single-ended and differential channels
  - Each channel can be configured to select any of the above analog inputs
- Sampling triggered by a task from software or a DPPI channel for full flexibility on sample frequency source from low-power 32.768 kHz RTC or more accurate 1/16 MHz timers
- One-shot conversion mode to sample a single channel
- Scans by taking samples of all selected channels in consequtive order
- 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
- On-the-fly limit checking

8.13.1 Overview

The ADC supports up to four external analog inputs. 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 four single-ended inputs, two differential inputs or a combination of these. Each channel can be configured to select:

- GPIO pins with analog input function, see Pin assignments on page 546, also marked with name **AIN**. Input range is 0 to VDD.
- **VDD** (divided down to a valid range using the programmable gain stage)
- **DVDD**

Channels can be sampled individually in one-shot or continuous sampling modes, or, using scan mode, multiple channels can be sampled in sequence. To improve noise performance, channels can be oversampled.

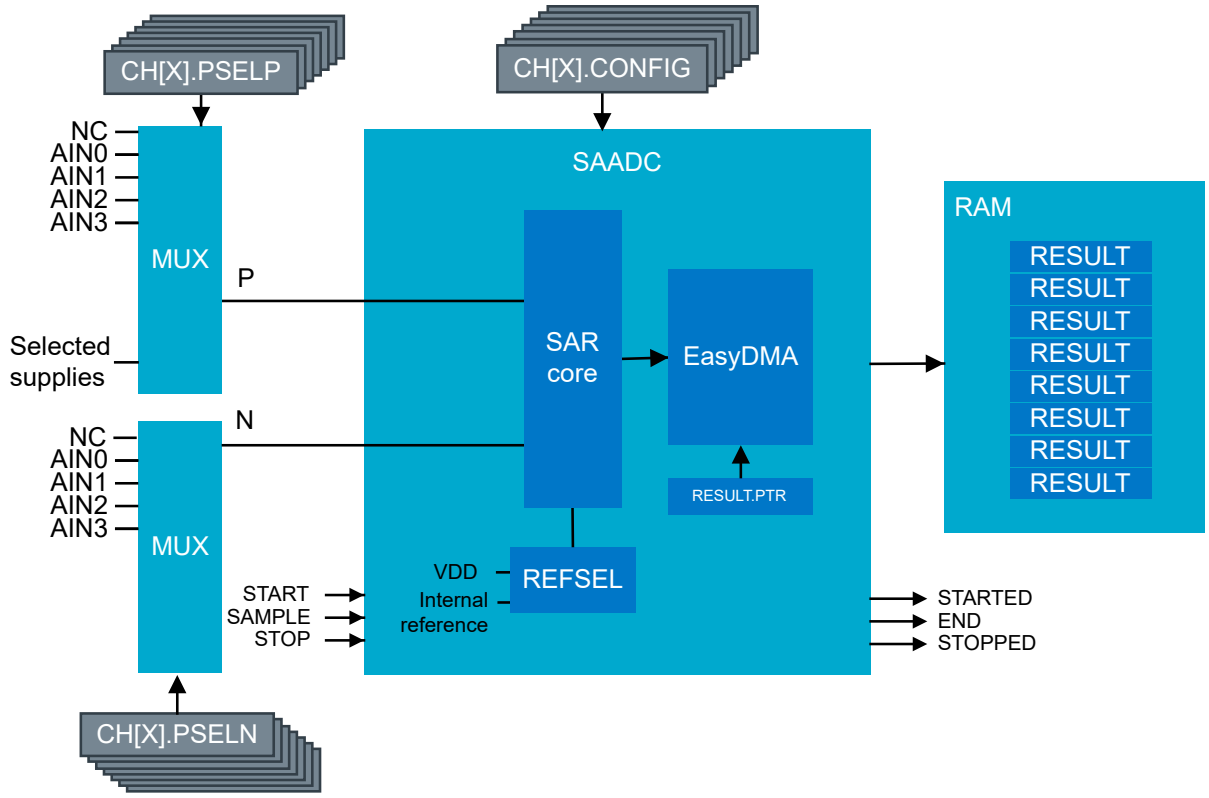


Figure 80: Simplified ADC block diagram

Internally, the ADC is always a differential analog-to-digital converter, but by default it is configured with single-ended input in the MODE field of the CH[n].CONFIG register. In single-ended mode, the negative input will be shorted to ground internally.

The assumption in single-ended mode is that the internal ground of the ADC is the same as the external ground that the measured voltage is referred to. The ADC is thus sensitive to ground bounce on the PCB in single-ended mode. If this is a concern we recommend using differential measurement.

8.13.2 Digital output

The output result of the ADC depends on the settings in the CH[n].CONFIG and RESOLUTION registers as follows:

$$RESULT = [V(P) - V(N)] * 1/REFERENCE * 2^{(RESOLUTION - m)}$$

where

V(P)

is the voltage at input P

V(N)

is the voltage at input N

m

is the mode setting. Use m=0 if CONFIG.MODE=SE, or m=1 if CONFIG.MODE=Diff

REFERENCE

is the selected reference voltage

The result generated by the ADC will deviate from the expected due to 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 temperature dependent offset. If the ADC is to operate over a large temperature range, we recommend running CALIBRATEOFFSET at regular intervals. The CALIBRATEDONE event will be generated when the calibration has been completed. Note that the DONE and RESULTDONE events will also be generated.

8.13.3 Analog inputs and channels

Up to four analog input channels, CH[n](n=0..3), can be configured.

See Shared resources for shared input with comparators.

Any one of the available channels can be enabled for the ADC to operate in one-shot mode. If more than one CH[n] is configured, the ADC enters scan mode.

An analog input is selected as a positive converter input if CH[n].PSELP is set, setting CH[n].PSELP also enables the particular channel.

An analog input is selected as a negative converter input if CH[n].PSELN is set. The CH[n].PSELN register will have no effect unless differential mode is enabled and CH[n].PSELP is set, see MODE field in CH[n].CONFIG register.

Important: It is not recommended to use the same analog input pin for multiple analog peripheral functions. See also Shared resources.

8.13.4 Operation modes

The ADC input configuration supports several modes of sampling.

- One-shot, one channel
- One-shot, scan (one sample for each channel)
- Continuous, one channel
- Continuous, scan

Note: Scan mode and oversampling should not be combined without burst.

1. The ADC must be enabled and started via the ENABLE register and START task.
2. At least one channel must be enabled (via the CH[n].CONFIG registers).
3. Now the ADC can be sampled, by triggering the SAMPLE task.

The ADC indicates a single ongoing conversion via the register STATUS. During scan, oversampling, or continuous modes, multiple conversions take place in the ADC. As a result, the value in STATUS register toggles at the end of each single conversion.

8.13.4.1 One-shot mode

One-shot operation is configured by enabling only one of the available channels defined by CH[n].PSELP, CH[n].PSELN, and CH[n].CONFIG registers.

Upon a SAMPLE task, the ADC powers up and starts to sample the input voltage. The CH[n].CONFIG.TACQ controls the acquisition time.

The time it takes to perform the first sample is $t_{PWRUP} + t_{ACQ} + t_{CONV}$, where t_{ACQ} is the acquisition time, t_{CONV} is the conversion time, and t_{PWRUP} is the time it takes to power up the ADC.

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual value has been transferred into RAM by EasyDMA. For more information, see EasyDMA on page 332.

8.13.4.2 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 fulfils the following criteria, depending on how many channels are active:

$$t_{SAMPLE} > t_{ACQ} + t_{CONV}$$

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.

A DONE event signals that one sample has been taken.

In this mode, the RESULTDONE event has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual value has been transferred into RAM by EasyDMA.

8.13.4.3 Improving sampling accuracy

Oversampling

Oversampling can improve the signal-to-noise ratio (SNR) by approximately 10log(OSR), where OSR is the oversampling ratio. Each oversampled result is obtained by acquiring $2^{OVERSAMPLE}$ samples and combining them in an accumulate-and-average filter.

Oversampling and scanning can only be combined with BURST mode enabled. Without BURST mode, oversampling and scanning will average across multiple input channels.

The OVERSAMPLE register controls the accumulator. $2^{OVERSAMPLE}$ samples must be taken before the result is written to RAM. This can be achieved by either of the following:

- Use the built-in SAADC local timer and the SAMPLERATE register to perform sampling.
- Use the TIMER peripheral and DPPI to trigger the SAADC SAMPLE task, to sample $2^{OVERSAMPLE}$ times at a fixed rate.
- Trigger the SAMPLE task $2^{OVERSAMPLE}$ times from software.
- Enable BURST mode and trigger the SAMPLE task once.

BURST mode can be enabled to avoid manually triggering the SAMPLE task $2^{OVERSAMPLE}$ times. When BURST is enabled, the ADC automatically samples the input $2^{OVERSAMPLE}$ times consecutively, with an approximate timing of $(t_{ACQ} + t_{CONV}) \times 2^{OVERSAMPLE}$. Apart from extending the conversion time, it otherwise behaves like one-shot mode.

A DONE event indicates that a single sample has been acquired, while a RESULTDONE event indicates that enough samples have been gathered to transfer an oversampled result to RAM.

Chopping

Chopping is a technique used to cancel offset voltages and other low frequency errors, and is enabled in the CH[n].CONFIG.CHOPPING register. When chopping is enabled, the input to the ADC is alternately inverted (chopped) at the input mux, and a conversion is performed for each phase of chop. This is

combined with oversampling, and ADC output is passed through the accumulate and average filter used in oversampling mode. Chopping is only avaible for slow channels.

8.13.4.4 Scan mode

A channel is considered enabled if CH[n].PSELP is set. If more than one channel, CH[n], is enabled, the ADC enters scan mode.

In scan mode, one SAMPLE task will trigger one conversion per enabled channel. The time it takes to sample all channels is:

```
Total time < Sum(CH[x].tACQ+tCONV), x=0..enabled channels
```

A DONE event signals that one channel has been sampled.

In this mode, the RESULTDONE event comes after each sample, that is, once for each channel.

8.13.4.5 High-speed channel

When the high-speed option is enabled, the the AIN0 (positive) and AIN1 (negative) pins can be used to sample signals with bandwidth up to 500 kHz.

High-speed mode is enabled by the HIGHSPEED field of the CH[n].CONFIG register.

In high-speed mode, AIN0 and AIN1 can be configured as either:

- A single-ended input (AIN0 vs GND), with AIN1 not connected.
- AIN0 as the positive side of a differential pair, with AIN1 as the negative input.

The other channels (AIN2 and AIN3) cannot be used in high-speed mode.

CHOPPING is not available in high-speed mode.

8.13.5 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 (in bytes) 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 333. 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.

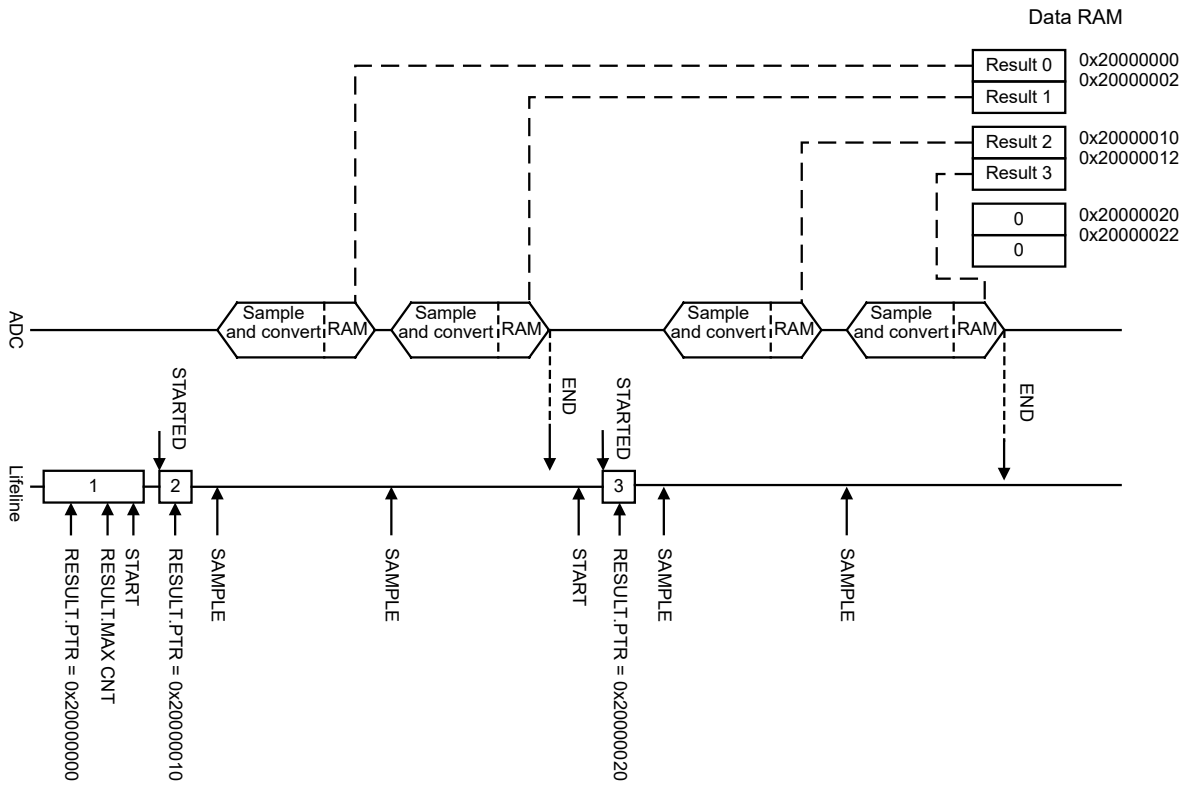


Figure 81: ADC

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 11 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 bytes 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.For more information about the scan mode, see Scan mode on page 332.

Note: Ensure the Result buffer can hold at least one result for each enabled channel by setting RESULT.MAXCNT ≥ 2 × (number of enabled channels). Each sample requires two bytes. Insufficient space leads to undefined behavior.

8.13.6 Reference

The ADC can use different reference voltages VREF, controlled in the REFSEL field of the CH[n].CONFIG register.

These are:

- Internal reference, VREF = 1.024V
- External reference (0.4 V up to VDD) , VREF provided by the EXTREF pin
- VDD

The AIN0-AIN3 inputs cannot exceed VDD, or be lower than VSS. The input ranges are also limited by the REFERENCE used. The condition

```
[V(P) - V(N) ] * 1/REFERENCE <= 1
```

must always hold true for valid measurements, otherwise the ADC will saturate and report the max value determined by the RESOLUTION.

8.13.7 Acquisition time

To sample the input voltage, the ADC connects a capacitor to the input.

For illustration, see the following figure. The acquisition time indicates how long the capacitor is connected, see TACQ field in CH[n].CONFIG register. The required acquisition time depends on the source (R_{source}) resistance. For high source resistance the acquisition time should be increased, see Acquisition time on page 334.

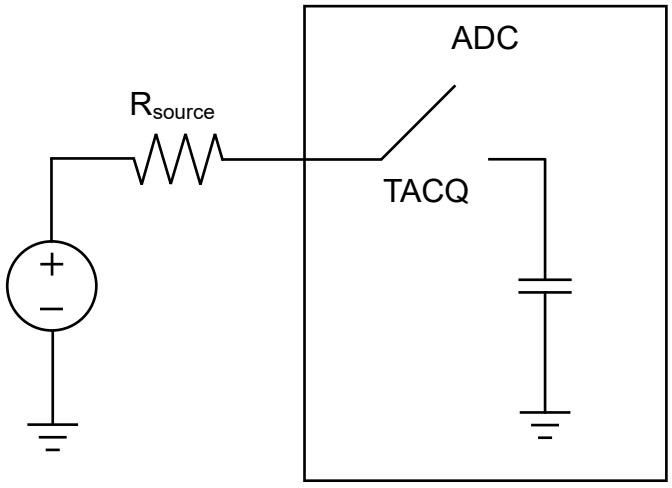


Figure 82: Simplified ADC sample network

TACQ	Maximum source resistance [kOhm]
1	<1
3	10
5	40
10	100
15	200
20	400
40	800

Table 39: Acquisition time

8.13.8 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 be generated.

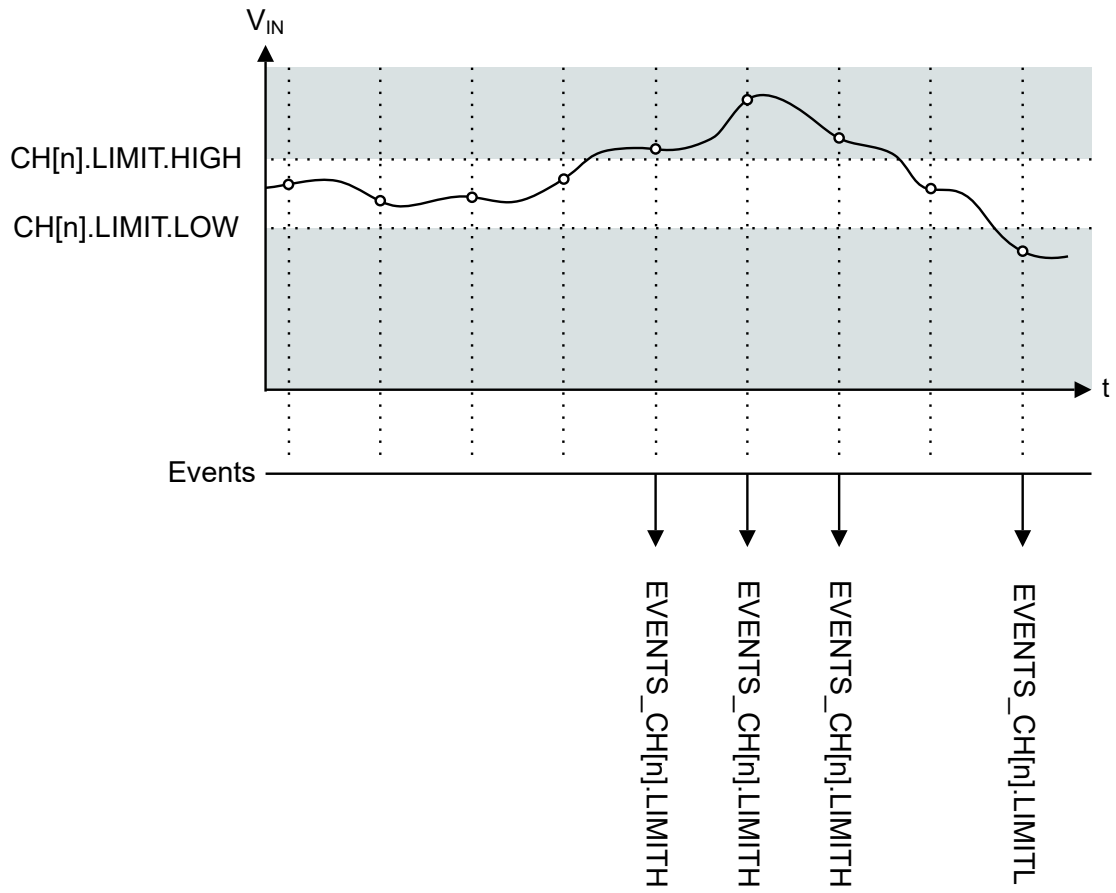


Figure 83: Example of limits monitoring on channel 'n'

Note that when setting the limits, CH[n].LIMIT.HIGH shall always be higher than or equal to CH[n].LIMIT.LOW. In other words, an event can be generated only when the input signal has been sampled outside of the defined limits. It is not possible to fire an event when the input signal is inside a defined range by swapping high and low limits.

The comparison to limits always takes place, there is no need to enable it. If comparison is not required on a channel, the software shall simply ignore the related events. In that situation, the value of the limits registers is irrelevant, so it does not matter if CH[n].LIMIT.LOW is lower than CH[n].LIMIT.HIGH or not.

8.13.9 Performance factors

Clock jitter, affecting sample timing accuracy, and circuit noise can affect ADC performance.

Jitter can be between START tasks or from START task to acquisition. START timer accuracy and startup times of regulators and references will contribute to variability. Sources of circuit noise may include CPU activity and the DC/DC regulator. Best ADC performance is achieved using START timing based on the TIMER module, HFXO clock source, and Constant Latency mode.

8.13.10 Registers

Instances

Instance	Domain	Base address	Description
SAADC	GLOBAL	0x400D5000	Successive approximation analog-to-digital converter SAADC

Configuration

Instance	Domain	Configuration
SAADC	GLOBAL	CURRENTAMOUNT register not included.

Register overview

Register	Offset	Description
TASKS_START	0x000	Start the ADC and prepare the result buffer in RAM
TASKS_SAMPLE	0x004	Take one ADC sample, if scan is enabled all channels are sampled. This task requires that SAADC has started, i.e. EVENTS_STARTED was set and EVENTS_STOPPED was not.
TASKS_STOP	0x008	Stop the ADC and terminate any on-going 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[m]
CH[n].PSELN	0x514	Input negative pin selection for CH[m]
CH[n].CONFIG	0x518	Input configuration for CH[m]
CH[n].LIMIT	0x51C	High/low limits for event monitoring a channel
BURST	0x5E8	Enable burst mode
RESOLUTION	0x5F0	Resolution configuration
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 bytes to transfer
RESULT.AMOUNT	0x634	Number of buffer bytes transferred since last START, updated after the END or STOPPED events
RESULT.CURRENTAMOUNT	0x638	Number of buffer bytes transferred since last START, continuously updated

8.13.10.1 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

8.13.10.2 TASKS_SAMPLE

Address offset: 0x004

Take one ADC sample, if scan is enabled all channels are sampled. This task requires that SAADC has started, i.e. EVENTS_STARTED was set and EVENTS_STOPPED was not.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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. This task requires that SAADC has started, i.e. EVENTS_STARTED was set and EVENTS_STOPPED was not.
			Trigger	1	Trigger task

8.13.10.3 TASKS_STOP

Address offset: 0x008

Stop the ADC and terminate any on-going 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 on-going conversion
			Trigger	1	Trigger task

8.13.10.4 TASKS_CALIBRATEOFFSET

Address offset: 0x00C

Starts offset auto-calibration

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

8.13.10.5 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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.13.10.6 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		[0..255]	DPPI channel that task SAMPLE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.13.10.7 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.13.10.8 SUBSCRIBE_CALIBRATEOFFSET

Address offset: 0x08C

Subscribe configuration for task CALIBRATEOFFSET

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that task CALIBRATEOFFSET will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.13.10.9 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

8.13.10.10 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

8.13.10.11 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

8.13.10.12 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																															
A	RW	EVENTS_RESULTDONE																																	
			NotGenerated	0																															
			Generated	1																															

8.13.10.13 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																															
A	RW	EVENTS_CALIBRATEDONE																																	
			NotGenerated	0																															
			Generated	1																															

8.13.10.14 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																															
A	RW	EVENTS_STOPPED																																	
			NotGenerated	0																															
			Generated	1																															

8.13.10.15 EVENTS_CH[n] (n=0..3)

Peripheral events.

8.13.10.15.1 EVENTS_CH[n].LIMITH (n=0..3)

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

8.13.10.15.2 EVENTS_CH[n].LIMITL (n=0..3)

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

8.13.10.16 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		[0..255]	DPPI channel that event STARTED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.17 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.18 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event DONE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.19 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event RESULTDONE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.20 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CALIBRATEDONE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.21 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.22 PUBLISH_CH[n] (n=0..3)

Publish configuration for events

8.13.10.22.1 PUBLISH_CH[n].LIMITH (n=0..3)

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CH[n].LIMITH will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.22.2 PUBLISH_CH[n].LIMITL (n=0..3)

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CH[n].LIMITL will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.13.10.23 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				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

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

ID	R/W	Field	Value ID	Value	Description
			Enabled	1	Enable
B	RW	END			Enable or disable interrupt for event END
			Disabled	0	Disable
			Enabled	1	Enable
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

8.13.10.24 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				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 W1S	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	DONE			Write '1' to enable interrupt for event DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	RESULTDONE			Write '1' to enable interrupt for event RESULTDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	CALIBRATEDONE			Write '1' to enable interrupt for event CALIBRATEDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	CH0LIMITH			Write '1' to enable interrupt for event CH0LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1S	CH0LIMITL			Write '1' to enable interrupt for event CH0LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1S	CH1LIMITH			Write '1' to enable interrupt for event CH1LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1S	CH1LIMITL			Write '1' to enable interrupt for event CH1LIMITL
			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				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
K	RW W1S	CH2LIMITH			Write '1' to enable interrupt for event CH2LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1S	CH2LIMITL			Write '1' to enable interrupt for event CH2LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	CH3LIMITH			Write '1' to enable interrupt for event CH3LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1S	CH3LIMITL			Write '1' to enable interrupt for event CH3LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.13.10.25 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				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 W1C	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	DONE			Write '1' to disable interrupt for event DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	RESULTDONE			Write '1' to disable interrupt for event RESULTDONE
			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				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
E	RW W1C	CALIBRATEDONE			Write '1' to disable interrupt for event CALIBRATEDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1C	CH0LIMITH			Write '1' to disable interrupt for event CH0LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1C	CH0LIMITL			Write '1' to disable interrupt for event CH0LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1C	CH1LIMITH			Write '1' to disable interrupt for event CH1LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1C	CH1LIMITL			Write '1' to disable interrupt for event CH1LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1C	CH2LIMITH			Write '1' to disable interrupt for event CH2LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	CH2LIMITL			Write '1' to disable interrupt for event CH2LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	CH3LIMITH			Write '1' to disable interrupt for event CH3LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	CH3LIMITL			Write '1' to disable interrupt for event CH3LIMITL
			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																						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																															
			Enabled	1																															
				Read: Enabled																															

8.13.10.26 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																															
A	R	STATUS																																	
			Ready	0																															
			Busy	1																															
				ADC is ready. No on-going conversion.																															
				ADC is busy. Single conversion in progress.																															

8.13.10.27 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																															
A	RW	ENABLE																																	
			Disabled	0																															
			Enabled	1																															
				Enable or disable ADC																															
				Disable ADC																															
				Enable ADC																															
				When enabled, the ADC will acquire access to the GPIO pins specified in the CH[n].PSELP and CH[n].PSELN registers.																															

8.13.10.28 CH[n].PSELP (n=0..3)

Address offset: 0x510 + (n × 0x10)

Input positive pin selection for CH[m]

The analog input is selected based on CH[m].PSELP.PORT and CH[m].PSELP.PIN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D C C C B B 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	RW	PIN			GPIO pin selection.
B	RW	PORT			GPIO port selection
C	RW	INTERNAL			Internal input selection for analog positive input when CH[m].PSELP.CONNECT = Internal
			Vdd	2	Connected to VDD, and divided by 8/30 (0.267)
			Vss	4	
D	RW	CONNECT			Connection
			NC	0	Not connected
			AnalogInput	1	Select analog input The analog input is connected based on CH[m].PSELP.PIN and CH[m].PSELP.PORT
			Internal	2	Selects internal inputs. The analog input is connected based on CH[m].PSELP.INTERNAL

8.13.10.29 CH[n].PSELN (n=0..3)

Address offset: 0x514 + (n × 0x10)

Input negative pin selection for CH[m]

The analog input is selected based on CH[m].PSELN.PORT and CH[m].PSELN.PIN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D C C C B B 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	RW	PIN			GPIO pin selection.
B	RW	PORT			GPIO Port selection
C	RW	INTERNAL			Internal input selection for Analog negative input when CH[m].PSELN.CONNECT = Internal
			Vdd	2	Connected to VDD, and divided by 8/30 (0.267)
			Vss	4	
D	RW	CONNECT			Connection
			NC	0	Not connected
			AnalogInput	1	Select analog input The analog input is connected based on CH[m].PSELN.PIN and CH[m].PSELN.PORT
			Internal	2	Selects internal inputs. The actual analog input is CH[m].PSELN.INTERNAL

8.13.10.30 CH[n].CONFIG (n=0..3)

Address offset: 0x518 + (n × 0x10)

Input configuration for CH[m]

Bit number				31 30 29 28 27 26 25 24 23 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 F F F F F F F F F E D D D C B A
Reset 0x00020000				**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	RW	CHOPPING			Enable chopping
			Disabled	0	Chopping is disabled
			Enabled	1	Chopping is enabled
B	RW	VREFPLPFDISABLE			Disable the internal low pass filter for the reference voltage Disabling the internal low pass filter enables a trade-off between faster reference voltage settling time and noise.
			Disabled	1	The internal low pass filter for the reference voltage is disabled (reference voltage is unfiltered and has faster settling time)
			Enabled	0	The internal low pass filter for the reference voltage is enabled (reference voltage is filtered)
C	RW	HIGHSPEED			Enable high speed inputs for the ADC, allowing the highest sampling bandwidth
			Disabled	0	High speed inputs are disabled
			Enabled	1	High speed inputs are enabled
D	RW	REFSEL			Reference control
			Internal	0	Internal 1.024 V reference
			External	1	External reference voltage
			Vdd	3	Reference given at VDD
E	RW	MODE			Enable differential mode
			SE	0	Single ended, PSELN will be ignored, negative input to ADC shorted to GND
			Diff	1	Differential
F	RW	TACQ		[3..319]	Acquisition time, the time the ADC uses to sample the input voltage. Resulting acquistion time is ((TACQ+1) x 125 ns)
G	RW	TCONV		[1..7]	Conversion time. Resulting conversion time is ((TCONV+1) x 250 ns)

8.13.10.31 CH[n].LIMIT (n=0..3)

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

8.13.10.32 BURST

Address offset: 0x5E8

Enable burst 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**
ID	R/W	Field	Value ID	Value	Description
A	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 and sends the average to Data RAM.

8.13.10.33 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

8.13.10.34 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

8.13.10.35 SAMPLERATE

Address offset: 0x5F8

Controls normal or continuous sample rate

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

8.13.10.36 RESULT

RESULT EasyDMA channel

8.13.10.36.1 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.13.10.36.2 RESULT.MAXCNT

Address offset: 0x630

Maximum number of buffer 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 A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 bytes to transfer

8.13.10.36.3 RESULT.AMOUNT

Address offset: 0x634

Number of buffer bytes transferred since last START, updated after the END or STOPPED events

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 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	AMOUNT			Number of buffer bytes transferred since last START, updated after the END or STOPPED events.

8.13.10.36.4 RESULT.CURRENTAMOUNT

Address offset: 0x638

Number of buffer bytes transferred since last START, continuously 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																		A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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 bytes transferred since last START, continuously updated.

8.14 SPIM — Serial peripheral interface controller with EasyDMA

The SPI controller peripheral (SPIM) with EasyDMA provides a full duplex, 4-wire synchronous serial communication interface.

The main features of SPIM are the following:

- EasyDMA direct transfer to and from RAM
- SPI mode [0..3]
- Individual selection of I/O pins
- Optional D/CX output line for distinguishing between command and data bytes
- Optional hardware controlled chip select (CSN)

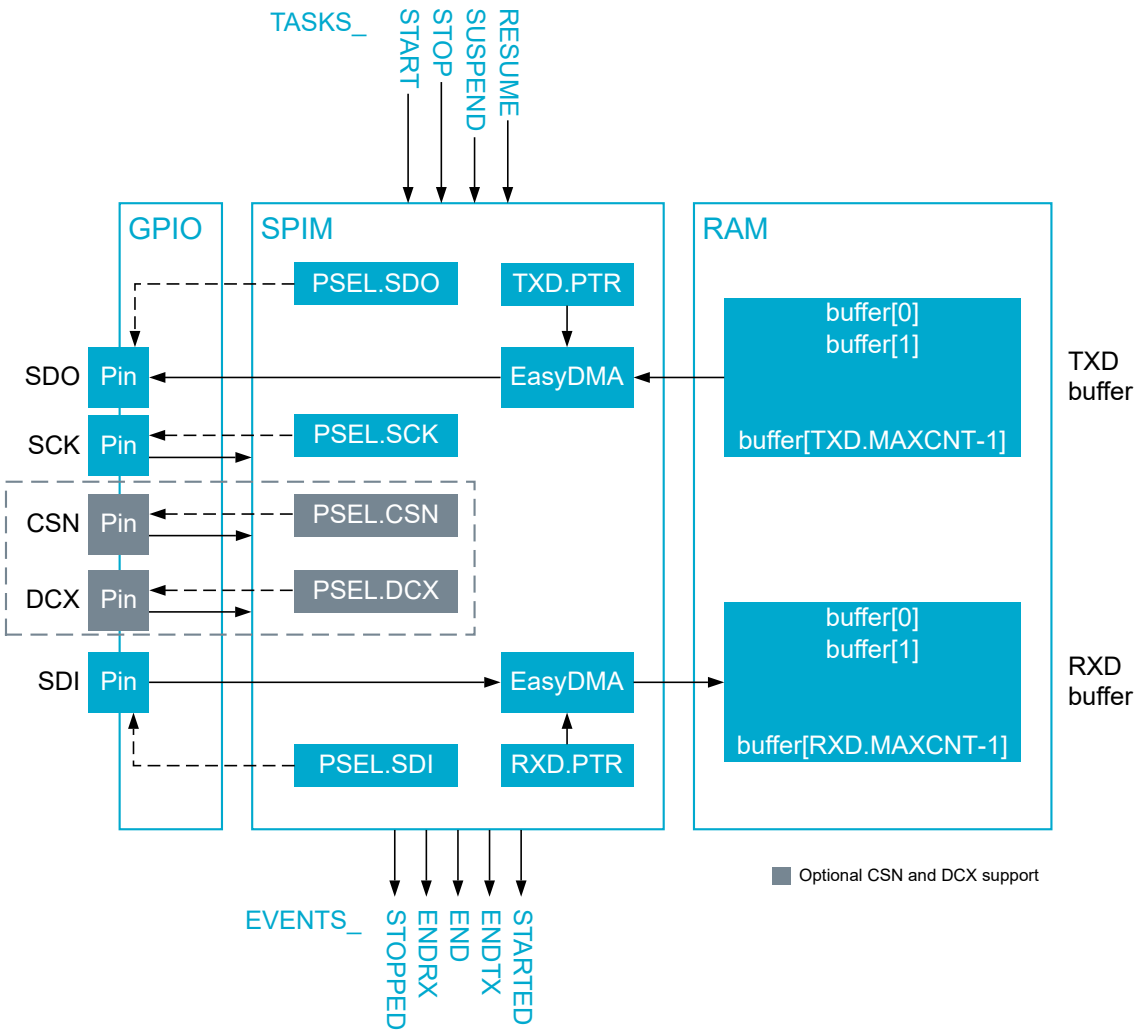


Figure 84: SPIM with EasyDMA

8.14.1 SPIM transaction sequence

An SPIM transaction is started by triggering the START task. This initiates a number of bytes to be transmitted/received on SDO/SDI.

The following figure illustrates an SPIM transaction.

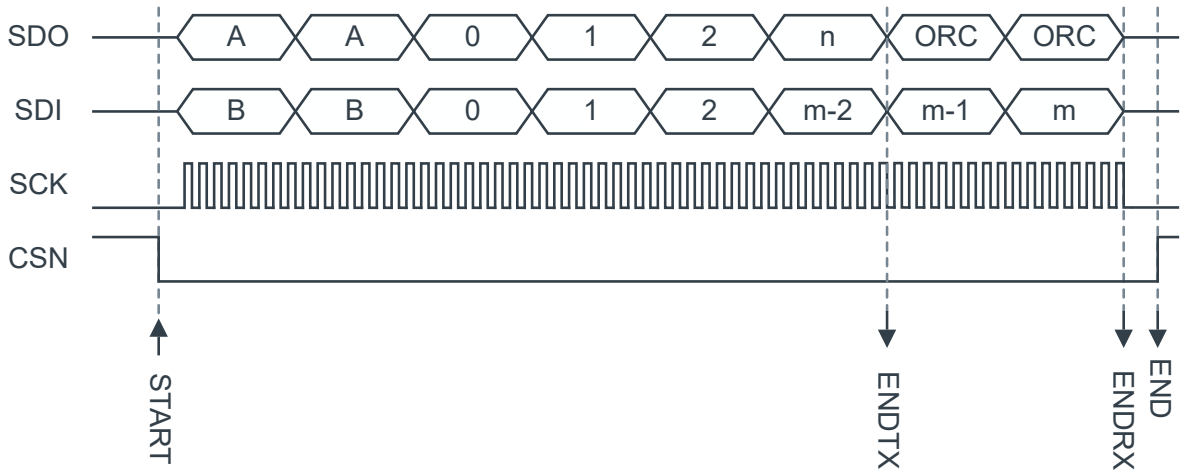


Figure 85: SPIM transaction

The ENDTX event is generated when all bytes in buffer DMA.TX.PTR on page 379 are transmitted. The number of bytes in the transmit buffer is specified in register DMA.TX.MAXCNT on page 379. The ENDRX event is generated when buffer DMA.RX.PTR on page 376 is full; that is when the number of bytes specified in register DMA.RX.MAXCNT on page 377 have been received. The transaction stops automatically after all bytes are transmitted or received. When the maximum number of bytes in the receive buffer is larger than the number of bytes in the transmit buffer, the contents of register ORC on page 375 will be transmitted after the last byte in the transmit buffer has been transmitted.

The END event is generated after both the ENDRX and ENDTX events have been generated.

SPIM is stopped by triggering the STOP task. A STOPPED event is generated when the SPIM has stopped. If the STOP task is triggered in the middle of a transaction, SPIM completes the process for 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 been generated when the SPIM peripheral stops, the ENDTX event will be generated, even if all bytes in the buffer have not been transmitted.

If the ENDRX event has not been generated when the SPIM stops, the ENDRX event will be generated even if the buffer DMA.RX.PTR on page 376 is not full.

A transaction can be suspended and resumed using the SUSPEND and RESUME tasks, respectively. When the SUSPEND task is triggered, SPIM completes transmitting and receiving the current byte before it is suspended.

8.14.2 D/CX functionality

Some SPI targets, such as display drivers, require an additional signal from the SPIM to distinguish between command and data bytes. This line is called D/CX.

SPIM supports additional signals such as a D/CX output line. The D/CX line is set low for transmitting command bytes and high for transmitting data bytes.

The D/CX pin number is selected using . The number of command bytes that precede the data bytes is configured using DCXCNT on page 374. Writing to DCXCNT on page 374 during an ongoing transmission is not allowed.

The following figure shows D/CX in use, where SPIM.DCXCNT=`1`.

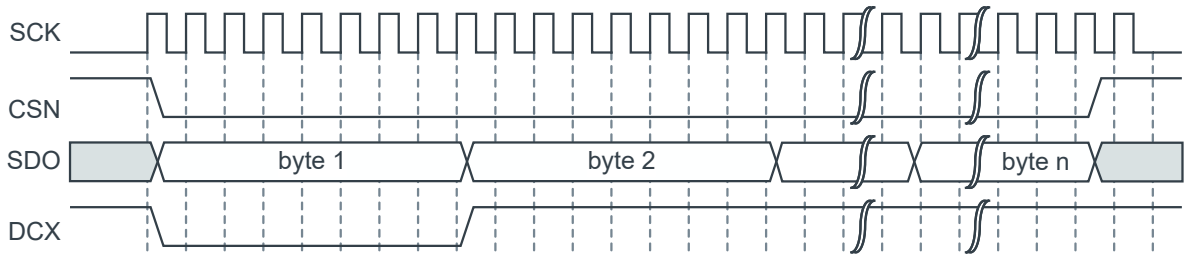


Figure 86: D/CX example

8.14.3 Chip select hardware control

To use CSN hardware control, set register PSEL.CSN on page 376 according to the Pin configuration on page 356.

When enabled, CSN is asserted automatically after TASKS_START on page 360 is triggered, and deasserted after EVENTS_END on page 364. The value in register IFTIMING.CSNDUR on page 374 sets the time between the falling edge of CSN and the first SCK edge. The same delay is used between the last SCK edge and the rising edge of CSN at the end of a transfer. It is also used between two transfers as the minimum CSN inactive time when the END_START short is enabled. The IFTIMING.CSNDUR is expressed in number of SPIM core clock periods.

The following figure shows a timing diagram with two SPI transmissions where the delay is controlled by IFTIMING.CSNDUR and is represented by t_{CSNDUR}.

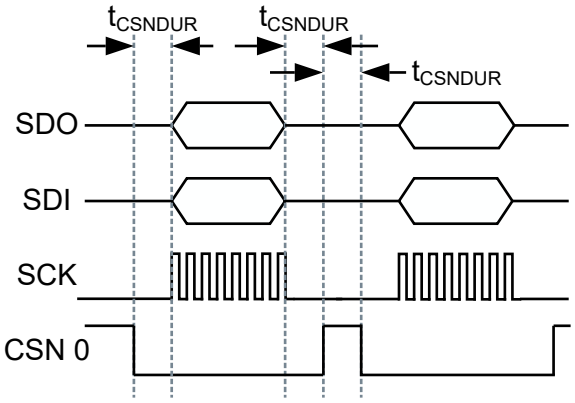


Figure 87: CSNDUR example

The following figure shows the timing delay for the combinations of polarity and phase settings. The register CSNPOL on page 374 determines the active polarity of the signal.

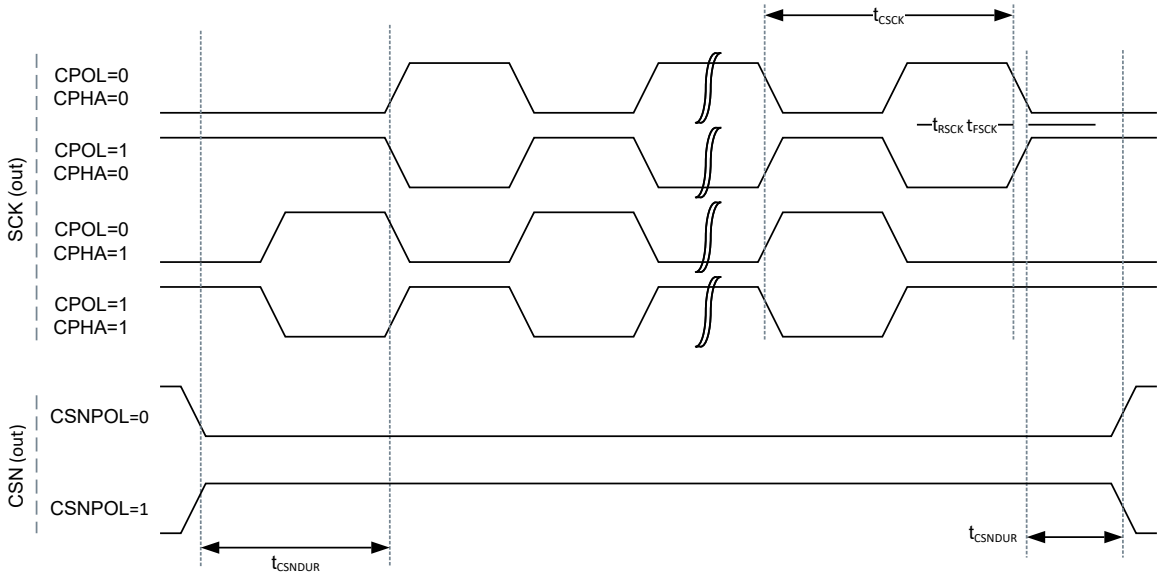


Figure 88: Polarity settings timing delay

8.14.4 Pin configuration

To configure pins for SPIM use, see the corresponding PSEL.n registers.

The contents of registers PSEL.SCK, PSEL.CSN, PSEL.DCX, PSEL.MOSI, and PSEL.MISO are only used when SPIM is enabled, and retained while the device is in System ON mode. The PSEL.n registers can be configured only when SPIM is disabled in register ENABLE on page 372.

To ensure correct behavior, the pins used by SPIM must be configured in the GPIO peripheral as described in GPIO configuration on page 357 before SPIM is enabled.

Only one peripheral can be assigned to drive a GPIO pin at a time. If more than one peripheral is assigned to a GPIO pin, it could result in unpredictable behavior.

SPIM signal	SPIM pin	Direction	Output value
SCK	As specified in PSEL.SCK on page 375	Output	Same as CONFIG.CPOL
CSN	As specified in PSEL.CSN on page 376	Output	Same as CONFIG.CPOL
DCX	As specified in PSEL.DCX on page 376	Output	1
SDO	As specified in PSEL.MOSI on page 375	Output	0
SDI	As specified in PSEL.MISO on page 376	Input	Not applicable

Table 40: GPIO configuration

SPIM supports SPI modes [0..3]. The clock polarity (CPOL) and the clock phase (CPHA) are configured in register CONFIG on page 373.

Mode	Clock polarity (CPOL)	Clock phase (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 41: SPI modes

8.14.5 Shared resources

The SPIM peripheral shares registers and other resources with peripherals that have the same ID as SPIM. Before SPIM can be configured and used, all peripherals that have the same ID as SPIM must be disabled.

Disabling a peripheral with the same ID as SPIM will not reset any shared SPIM registers. Configure all SPIM registers to ensure they operate correctly.

See the Instantiation table in Instantiation on page 141 for details on peripherals and their IDs.

8.14.6 EasyDMA

SPIM uses EasyDMA to fetch data to transmit from RAM or store received data in RAM.

SPIM implements the following EasyDMA channels.

Channel	Type	Register Cluster
TXD	READER	DMA.TX.PTR on page 379
RXD	WRITER	DMA.RX.PTR on page 376

Table 42: SPIM EasyDMA channels

The .PTR and .MAXCNT registers are double-buffered. After receiving the STARTED event, the registers can be written to before the next transmission.

SPIM automatically stops 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 RX.END and TX.END events indicate that EasyDMA has finished accessing buffers in RAM. Both RX and TX must be finished before the END event is generated.

If several AHB bus masters try to access the same AHB slave at the same time, AHB bus congestion can occur. In this case, the EasyDMA channel behavior will depend on the SPIM instance. Refer to Instances on page 358 for information about what behavior is expected in each instance.

See EasyDMA for more detailed information.

8.14.7 Low power

When the peripheral is not needed, stop and disable SPIM to ensure lowest possible power consumption.

When the STOP task is sent, the software must wait until the STOPPED event is received before disabling the peripheral through the ENABLE register. If the peripheral is already stopped, the STOP task is not needed to ensure data is not lost.

8.14.8 Registers

Instances

Instance	Domain	Base address	Description
SPIM20	GLOBAL	0x400C6000	SPI controller SPIM20
SPIM21	GLOBAL	0x400C7000	SPI controller SPIM21
SPIM22	GLOBAL	0x400C8000	SPI controller SPIM22

Configuration

Instance	Domain	Configuration
SPIM20	GLOBAL	Use GPIO port P1 CSN functionality is supported. DCX functionality is supported. Peripheral core frequency is 16 MHz. Prescaler divisor range is 2..126
SPIM21	GLOBAL	Use GPIO port P1 CSN functionality is supported. DCX functionality is supported. Peripheral core frequency is 16 MHz. Prescaler divisor range is 2..126
SPIM22	GLOBAL	Use GPIO port P1 CSN functionality is supported. DCX functionality is supported. Peripheral core frequency is 16 MHz. Prescaler divisor range is 2..126

Register overview

Register	Offset	Description
TASKS_START	0x000	Start SPI transaction
TASKS_STOP	0x004	Stop SPI transaction
TASKS_SUSPEND	0x00C	Suspend SPI transaction
TASKS_RESUME	0x010	Resume SPI transaction
TASKS_DMA.RX.ENABLEMATCH[n]	0x030	Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.RX.DISABLEMATCH[n]	0x040	Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
SUBSCRIBE_START	0x080	Subscribe configuration for task START
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x08C	Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x090	Subscribe configuration for task RESUME
SUBSCRIBE_DMA.RX.ENABLEMATCH[n]	0x0B0	Subscribe configuration for task ENABLEMATCH[n]
SUBSCRIBE_DMA.RX.DISABLEMATCH[n]	0x0C0	Subscribe configuration for task DISABLEMATCH[n]
EVENTS_STARTED	0x100	SPI transaction has started
EVENTS_STOPPED	0x104	SPI transaction has stopped
EVENTS_END	0x108	End of RXD buffer and TXD buffer reached
EVENTS_DMA.RX.END	0x14C	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.RX.READY	0x150	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.RX.BUSERROR	0x154	An error occured during the bus transfer.
EVENTS_DMA.RX.MATCH[n]	0x158	Pattern match is detected on the DMA data bus.
EVENTS_DMA.TX.END	0x168	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.TX.READY	0x16C	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.TX.BUSERROR	0x170	An error occured during the bus transfer.
PUBLISH_STARTED	0x180	Publish configuration for event STARTED
PUBLISH_STOPPED	0x184	Publish configuration for event STOPPED
PUBLISH_END	0x188	Publish configuration for event END
PUBLISH_DMA.RX.END	0x1CC	Publish configuration for event END
PUBLISH_DMA.RX.READY	0x1D0	Publish configuration for event READY
PUBLISH_DMA.RX.BUSERROR	0x1D4	Publish configuration for event BUSERROR
PUBLISH_DMA.RX.MATCH[n]	0x1D8	Publish configuration for event MATCH[n]
PUBLISH_DMA.TX.END	0x1E8	Publish configuration for event END
PUBLISH_DMA.TX.READY	0x1EC	Publish configuration for event READY
PUBLISH_DMA.TX.BUSERROR	0x1F0	Publish configuration for event BUSERROR
SHORTS	0x200	Shortcuts between local events and tasks
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable SPIM
PRESCALER	0x52C	The prescaler is used to set the SPI frequency.
CONFIG	0x554	Configuration register
IFTIMING.RXDELAY	0x5AC	Sample delay for input serial data on SDI
IFTIMING.CSNDUR	0x5B0	Minimum duration between edge of CSN and edge of SCK. When SHORTS.END_START is used, this is also the minimum duration CSN must stay high between transactions.
DCXCNT	0x5B4	DCX configuration
CSNPOL	0x5B8	Polarity of CSN output
ORC	0x5C0	Byte transmitted after TXD.MAXCNT bytes have been transmitted in the case when RXD.MAXCNT is greater than TXD.MAXCNT
PSEL.SCK	0x600	Pin select for SCK
PSEL.MOSI	0x604	Pin select for SDO signal
PSEL.MISO	0x608	Pin select for SDI signal
PSEL.DCX	0x60C	Pin select for DCX signal

Register	Offset	Description
PSEL.CSN	0x610	Pin select for CSN
DMA.RX.PTR	0x704	RAM buffer start address
DMA.RX.MAXCNT	0x708	Maximum number of bytes in channel buffer
DMA.RX.AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.RX.LIST	0x714	EasyDMA list type
DMA.RX.TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.RX.BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.
DMA.RX.MATCH.CONFIG	0x724	Configure individual match events
DMA.RX.MATCH.CANDIDATE[n]	0x728	The data to look for - any match will trigger the MATCH[n] event, if enabled.
DMA.TX.PTR	0x73C	RAM buffer start address
DMA.TX.MAXCNT	0x740	Maximum number of bytes in channel buffer
DMA.TX.AMOUNT	0x744	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.TX.LIST	0x74C	EasyDMA list type
DMA.TX.TERMINATEONBUSERROR	0x754	Terminate the transaction if a BUSERROR event is detected.
DMA.TX.BUSERRORADDRESS	0x758	Address of transaction that generated the last BUSERROR event.

8.14.8.1 TASKS_START

Address offset: 0x000

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																															
A	W	TASKS_START																																	
			Trigger	1																															

8.14.8.2 TASKS_STOP

Address offset: 0x004

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																															
A	W	TASKS_STOP																																	
			Trigger	1																															

8.14.8.3 TASKS_SUSPEND

Address offset: 0x00C

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

8.14.8.4 TASKS_RESUME

Address offset: 0x010

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

8.14.8.5 TASKS_DMA

Peripheral tasks.

8.14.8.5.1 TASKS_DMA.RX

Peripheral tasks.

8.14.8.5.1.1 TASKS_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x030 + (n × 0x4)

Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG 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	ENABLEMATCH			Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.14.8.5.1.2 TASKS_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG 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	DISABLEMATCH			Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.14.8.6 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.7 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.8 SUBSCRIBE_SUSPEND

Address offset: 0x08C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.9 SUBSCRIBE_RESUME

Address offset: 0x090

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.10 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.14.8.10.1 SUBSCRIBE_DMA.RX

Subscribe configuration for tasks

8.14.8.10.1.1 SUBSCRIBE_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task ENABLEMATCH[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		[0..255]	DPPI channel that task ENABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.10.1.2 SUBSCRIBE_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task DISABLEMATCH[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		[0..255]	DPPI channel that task DISABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.14.8.11 EVENTS_STARTED

Address offset: 0x100

SPI transaction 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			SPI transaction has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.12 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

8.14.8.13 EVENTS_END

Address offset: 0x108

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

8.14.8.14 EVENTS_DMA

Peripheral events.

8.14.8.14.1 EVENTS_DMA.RX

Peripheral events.

8.14.8.14.1.1 EVENTS_DMA.RX.END

Address offset: 0x14C

Generated after all MAXCNT bytes have been transferred

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated after all MAXCNT bytes have been transferred
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.1.2 EVENTS_DMA.RX.READY

Address offset: 0x150

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.1.3 EVENTS_DMA.RX.BUSERROR

Address offset: 0x154

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.1.4 EVENTS_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x158 + (n × 0x4)

Pattern match is detected on the DMA data bus.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MATCH			Pattern match is detected on the DMA data bus.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.2 EVENTS_DMA.TX

Peripheral events.

8.14.8.14.2.1 EVENTS_DMA.TX.END

Address offset: 0x168

Generated after all MAXCNT bytes have been transferred

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated after all MAXCNT bytes have been transferred
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.2.2 EVENTS_DMA.TX.READY

Address offset: 0x16C

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.14.2.3 EVENTS_DMA.TX.BUSERROR

Address offset: 0x170

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.14.8.15 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		[0..255]	DPPI channel that event STARTED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.16 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		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.17 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.18 PUBLISH_DMA

Publish configuration for events

8.14.8.18.1 PUBLISH_DMA.RX

Publish configuration for events

8.14.8.18.1.1 PUBLISH_DMA.RX.END

Address offset: 0x1CC

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.18.1.2 PUBLISH_DMA.RX.READY

Address offset: 0x1D0

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.18.1.3 PUBLISH_DMA.RX.BUSERROR

Address offset: 0x1D4

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.18.1.4 PUBLISH_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x1D8 + (n × 0x4)

Publish configuration for event MATCH[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		[0..255]	DPPI channel that event MATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.14.8.18.2 PUBLISH_DMA.TX

Publish configuration for events

8.14.8.18.2.1 PUBLISH_DMA.TX.END

Address offset: 0x1E8

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																															
A	RW	CHIDX		[0..255]																								DPPI channel that event END will publish to								
B	RW	EN																																	
			Disabled	0																								Disable publishing								
			Enabled	1																								Enable publishing								

8.14.8.18.2.2 PUBLISH_DMA.TX.READY

Address offset: 0x1EC

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																															
A	RW	CHIDX		[0..255]																								DPPI channel that event READY will publish to								
B	RW	EN																																	
			Disabled	0																								Disable publishing								
			Enabled	1																								Enable publishing								

8.14.8.18.2.3 PUBLISH_DMA.TX.BUSERROR

Address offset: 0x1F0

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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																															
A	RW	CHIDX		[0..255]																								DPPI channel that event BUSERROR will publish to								
B	RW	EN																																	
			Disabled	0																								Disable publishing								
			Enabled	1																								Enable publishing								

8.14.8.19 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				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	END_START			Shortcut between event END and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B-E	RW	DMA_RX_MATCH[i]_DMA_RX_ENABLEMA +1)%4] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.ENABLEMATCH[(i+1)%4] Allows daisy-chaining match events.
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F-I	RW	DMA_RX_MATCH[i]_DMA_RX_DISABLEMATCH[i] (i=0..3)			Shortcut between event DMA.RX.MATCH[i] and task DMA.RX.DISABLEMATCH[i]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.14.8.20 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				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 W1S	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	DMARXEND			Write '1' to enable interrupt for event DMARXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	DMARXREADY			Write '1' to enable interrupt for event DMARXREADY
			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				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
F	RW W1S	DMARXBUSERROR			Write '1' to enable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G-J	RW W1S	DMARXMATCH[i] (i=0..3)			Write '1' to enable interrupt for event DMARXMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1S	DMATXEND			Write '1' to enable interrupt for event DMATXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1S	DMATXREADY			Write '1' to enable interrupt for event DMATXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	DMATXBUSERROR			Write '1' to enable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.14.8.21 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				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 W1C	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	END			Write '1' to disable interrupt 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				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
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	DMARXEND			Write '1' to disable interrupt for event DMARXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	DMARXREADY			Write '1' to disable interrupt for event DMARXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	DMARXBUSERROR			Write '1' to disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G-J	RW W1C	DMARXMATCH[i] (i=0..3)			Write '1' to disable interrupt for event DMARXMATCH[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1C	DMATXEND			Write '1' to disable interrupt for event DMATXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	DMATXREADY			Write '1' to disable interrupt for event DMATXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	DMATXBUSERROR			Write '1' to disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.14.8.22 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

8.14.8.23 PRESCALER

Address offset: 0x52C

The prescaler is used to set the SPI frequency.

The prescaler divides the core clock by the divisor to make the SPI clock. The resulting frequency is given by 'core clock' / DIVISOR. Different instances of the SPIM might have different core clocks. The SPIM core clock and divisor limits is given in the instance table in Instances on page 358.

Note that a low prescaler setting may require changing the default RXDELAY value to ensure correct sampling.

Only even numbers is allowed for the divisor.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 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 0 1 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	DIVISOR		2..126	Core clock to SCK divisor

8.14.8.24 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

8.14.8.25 IFTIMING.RXDELAY

Address offset: 0x5AC

Sample delay for input serial data on SDI

If the value is written larger than the maximum value, the maximum value will 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
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 SDI. The value specifies the number of SPIM core clock cycles 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.

8.14.8.26 IFTIMING.CSNDUR

Address offset: 0x5B0

Minimum duration between edge of CSN and edge of SCK. When SHORTS.END_START is used, this is also the 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. When SHORTS.END_START is used, this is the minimum duration CSN must stay high between transactions. The value is specified in number of SPIM core clock cycles. Note that for low values of CSNDUR, the system turnaround time will dominate the actual time between transactions.

8.14.8.27 DCXCNT

Address offset: 0x5B4

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.

8.14.8.28 CSNPOL

Address offset: 0x5B8

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**
ID	R/W	Field	Value ID	Value	Description
A	RW	CSNPOL[i] (i=0..0)			Polarity of CSN output
			LOW	0	Active low (idle state high)
			HIGH	1	Active high (idle state low)

8.14.8.29 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**
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.

8.14.8.30 PSEL.SCK

Address offset: 0x600

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 B B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.14.8.31 PSEL.MOSI

Address offset: 0x604

Pin select for SDO 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 B B A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.14.8.32 PSEL.MISO

Address offset: 0x608

Pin select for SDI signal

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												B	B	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..7]																								Port number							
C	RW	CONNECT																										Connection							
			Disconnected	1																								Disconnect							
			Connected	0																								Connect							

8.14.8.33 PSEL.DCX

Address offset: 0x60C

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																												B	B	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..7]																								Port number							
C	RW	CONNECT																										Connection							
			Disconnected	1																								Disconnect							
			Connected	0																								Connect							

8.14.8.34 PSEL.CSN

Address offset: 0x610

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																												B	B	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..7]																								Port number							
C	RW	CONNECT																										Connection							
			Disconnected	1																								Disconnect							
			Connected	0																								Connect							

8.14.8.35 DMA.RX.PTR

Address offset: 0x704

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.14.8.36 DMA.RX.MAXCNT

Address offset: 0x708

Maximum number of bytes in channel 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 channel buffer

8.14.8.37 DMA.RX.AMOUNT

Address offset: 0x70C

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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.

8.14.8.38 DMA.RX.LIST

Address offset: 0x714

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	TYPE			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

8.14.8.39 DMA.RX.TERMINATEONBUSERROR

Address offset: 0x71C

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.14.8.40 DMA.RX.BUSERRORADDRESS

Address offset: 0x720

Address of transaction that generated the last BUSERROR 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 A A A A A A A A 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	ADDRESS			

8.14.8.41 DMA.RX.MATCH

Registers to control the behavior of the pattern matcher engine

8.14.8.41.1 DMA.RX.MATCH.CONFIG

Address offset: 0x724

Configure individual match events

Bit number				31 30 29 28 27 26 25 24 23 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-D	RW	ENABLE[i] (i=0..3)			Enable match filter i
			Disabled	0	Match filter disabled
			Enabled	1	Match filter enabled
E-H	RW	ONESHOT[i] (i=0..3)			Configure match filter i as one-shot or sticky One-shot match filters can be used together with shortcuts to check for continuous data sequences by disabling the filter if the next data is not a match. **Note:** The presence of these shorts depends on the configuration of the peripheral integrating this EasyDMA.
			Continuous	0	Match filter stays enabled until disabled by task
			Oneshot	1	Match filter stays enabled until next data word is received

8.14.8.41.2 DMA.RX.MATCH.CANDIDATE[n] (n=0..3)

Address offset: 0x728 + (n × 0x4)

The data to look for - any match will trigger the MATCH[n] event, if enabled.

Note: This register can be updated while a transfer is in progress, but the new value will not take effect until a match has been found or the transfer is done. That makes it possible to write a new set of match words which will be searched for immediately after the event triggers.

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

8.14.8.42 DMA.TX.PTR

Address offset: 0x73C

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address. **Note:** See the memory chapter for details about which memories are available for EasyDMA.

8.14.8.43 DMA.TX.MAXCNT

Address offset: 0x740

Maximum number of bytes in channel 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 channel buffer

8.14.8.44 DMA.TX.AMOUNT

Address offset: 0x744

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.14.8.45 DMA.TX.LIST

Address offset: 0x74C

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																															
A	RW	TYPE																																	
			Disabled	0																															
			ArrayList	1																															

8.14.8.46 DMA.TX.TERMINATEONBUSERROR

Address offset: 0x754

Terminate the transaction if a BUSERROR event is 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																															
A	RW	ENABLE																																	
			Disabled	0																															
			Enabled	1																															

8.14.8.47 DMA.TX.BUSERRORADDRESS

Address offset: 0x758

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	R	ADDRESS																																	

8.15 SPIS — Serial peripheral interface target with EasyDMA

The SPI target peripheral (SPIS) with EasyDMA provides a full duplex, 4-wire synchronous serial communication interface.

The main features of SPIS are the following:

- EasyDMA direct transfer to and from RAM
- SPI mode [0..3]
- Individual selection of I/O pins
- Hardware-based semaphore mechanisms for synchronizing access to data buffers by SPIS and CPU

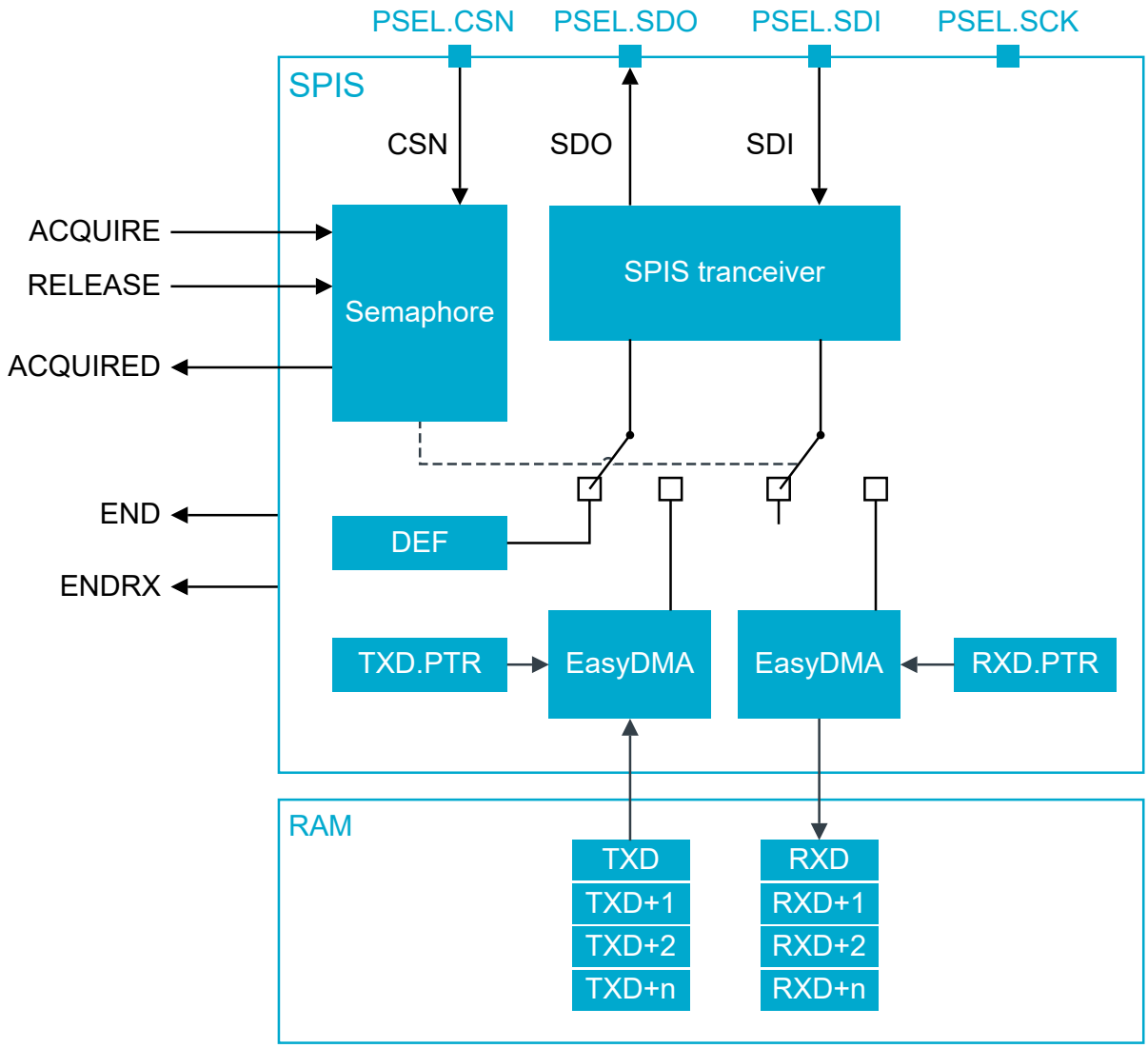


Figure 89: SPIS

8.15.1 SPI modes

SPIS supports SPI modes [0..3]. Modes CPOL and CPHA are set in the CONFIG register.

Mode	Clock polarity (CPOL)	Clock phase (CPHA)
SPI_MODE0	0 (Active `High`)	0 (Sample on Leading)
SPI_MODE1	0 (Active `High`)	1 (Sample on Trailing)
SPI_MODE2	1 (Active `Low`)	0 (Sample on Leading)
SPI_MODE3	1 (Active `Low`)	1 (Sample on Trailing)

Table 43: SPI modes

8.15.2 Shared resources

The SPIS peripheral shares registers and other resources with peripherals that have the same ID as SPIS. Before SPIS can be configured and used, all peripherals that have the same ID as SPIS must be disabled.

Disabling a peripheral with the same ID as SPIS will not reset any shared SPIS registers. Configure all SPIS registers to ensure they operate correctly.

See the Instantiation table in Instantiation on page 141 for details on peripherals and their IDs.

8.15.3 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 44: SPIS EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 19.

If RXD.MAXCNT is greater than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register.

The END event indicates that EasyDMA is finished accessing the RAM buffer.

8.15.4 SPIS operation

SPIS uses two memory pointers. RXD.PTR points to the RXD buffer (receive buffer) and TXD.PTR points to the TXD buffer (transmit buffer). Because these buffers are located in RAM, which can be accessed by both SPIS and the CPU, a hardware based semaphore mechanism is implemented to enable safe sharing.

The CPU must acquire the SPI semaphore before it can safely update the RXD.PTR and TXD.PTR pointers. The ACQUIRE task must be triggered for the CPU to receive the ACQUIRED event and have access to the semaphore. When the CPU has updated the RXD.PTR and TXD.PTR pointers, the CPU must release the semaphore before SPIS can acquire it.

The CPU releases the semaphore by triggering the RELEASE task, as illustrated in the following figure. Triggering the RELEASE task when the CPU does not have access to the semaphore will have no effect. See Semaphore operation on page 384 for more information.

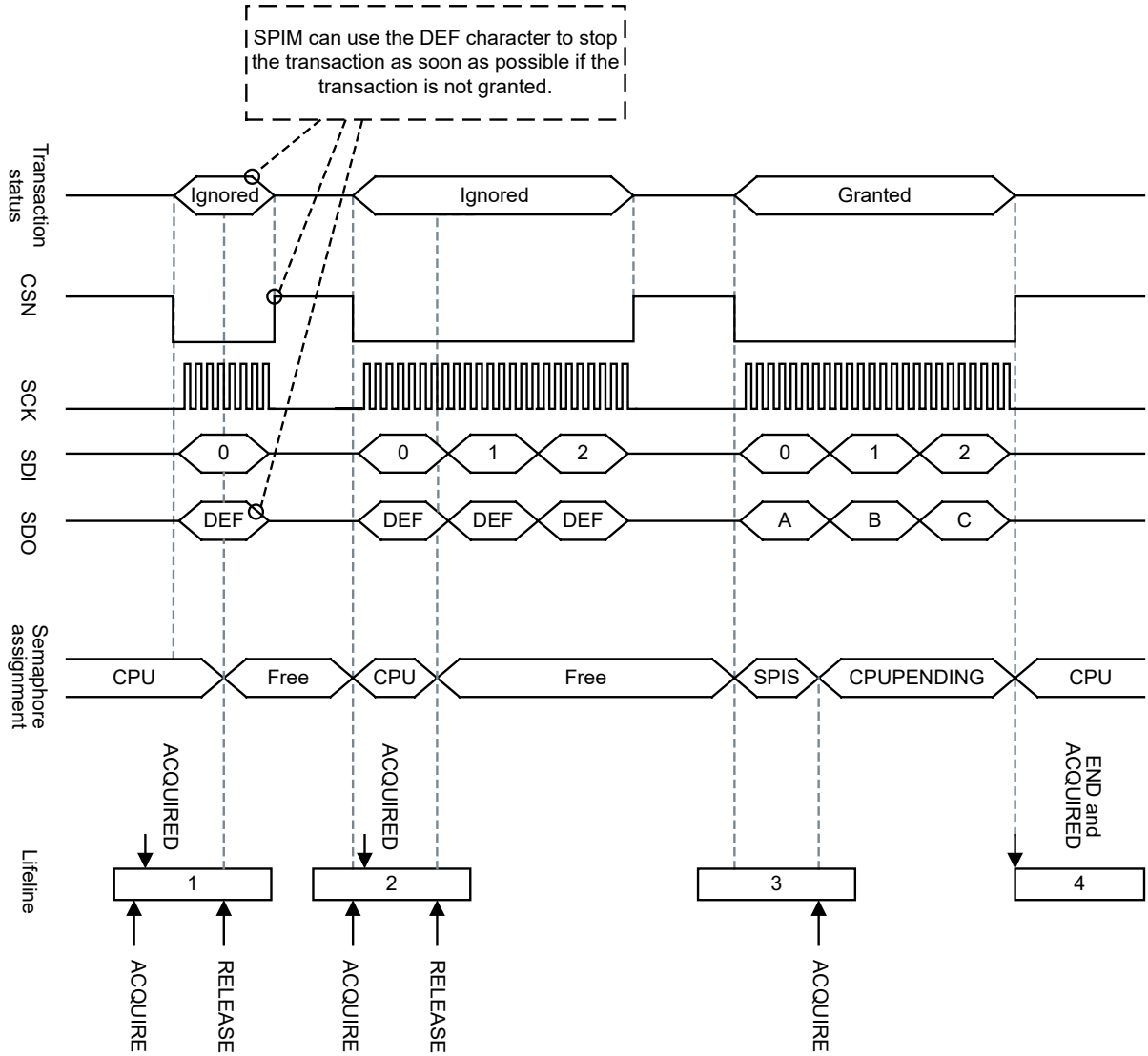


Figure 90: SPI transaction when shortcut between END and ACQUIRE is enabled

If the CPU is not able to reconfigure 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. This enables the CPU 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 SPIS can receive in one granted transaction. If SPIS 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 SPIS can transmit in one granted transaction. If SPIS 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 complete. The TXD.AMOUNT register indicates how many bytes were read from the TX buffer in the last transaction. 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.

8.15.5 Semaphore operation

The semaphore is a mechanism implemented inside the SPIS peripheral that prevents SPIS and CPU from accessing data buffers simultaneously.

By default, the semaphore is assigned to the CPU after the SPIS peripheral is enabled. An ACQUIRED event will not be generated for this initial semaphore handover. If the ACQUIRE task is triggered while the semaphore is assigned to the CPU, an ACQUIRED event will be generated immediately. The following figure illustrates the transitions between states in the semaphore based on the relevant tasks and events.

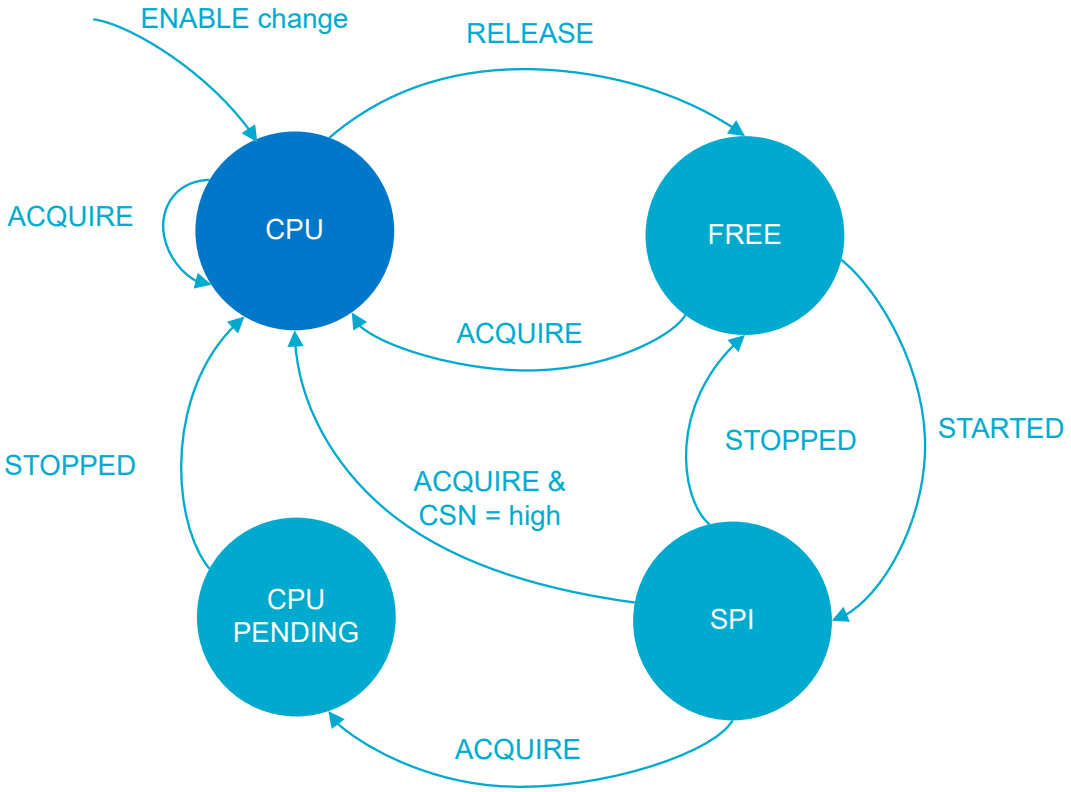


Figure 91: SPI semaphore FSM

Note: The semaphore mechanism does not prevent the CPU from performing read or write access to the RXD.PTR register, TXD.PTR registers, or 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.

SPIS will try to acquire the semaphore when the STARTED event is detected. If SPIS does not obtain the semaphore, the transaction will be ignored and the semaphore is retained by the CPU. All incoming data on SDI will be discarded and the DEF (default) character will be clocked out on the SDO line throughout the transaction. This is also true if the semaphore is released by the CPU during the transaction. If a race condition occurs where the CPU and SPIS 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 383, the CPU is given the semaphore.

If SPIS acquires the semaphore, the transaction will be granted. The incoming data on SDI will be stored in the RXD buffer and the data in the TXD buffer will be clocked out on SDO.

When a transaction is complete and CSN goes `HIGH`, SPIS will automatically release the semaphore and generate the END event.

SPIS can be granted multiple transactions in a row as long as the semaphore is available.

If the CPU tries to acquire the semaphore while it is assigned to SPIS, an immediate handover will not be granted. After the granted transaction is complete, SPIS releases the semaphore to the CPU. 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.

8.15.6 Pin configuration

The CSN, SCK, SDI, and SDO signals associated with SPIS 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 is set to `Disconnected`, the associated SPIS signal will not be connected to any physical pins.

These registers and their configurations are only used when SPIS is enabled, and retained as long as the device is in System ON mode. See POWER — Power control on page 79 for more information about power modes. When the peripheral is disabled, the pins behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. Only configure PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO when SPIS is disabled.

Before enabling SPIS, the pins used by SPIS must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 385. This ensures that the pins are driven correctly if SPIS becomes temporarily disabled, or if the device enters System OFF mode. This configuration must be retained in the GPIO for the selected pins to be recognized by an external SPI controller.

The SDO line is set `HIGH` as long as SPIS 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	
SDI	As specified in PSEL.MOSI	Input	Not applicable	
SDO	As specified in PSEL.MISO	Input	Not applicable	Emulates that SPIS is not selected.

Table 45: GPIO configuration before enabling peripheral

8.15.7 Registers

Instances

Instance	Domain	Base address	Description
SPIS20	GLOBAL	0x400C6000	SPI peripheral SPIS20
SPIS21	GLOBAL	0x400C7000	SPI peripheral SPIS21
SPIS22	GLOBAL	0x400C8000	SPI peripheral SPIS22

Configuration

Instance	Domain	Configuration
SPIS20	GLOBAL	Use GPIO port P1
SPIS21	GLOBAL	Use GPIO port P1
SPIS22	GLOBAL	Use GPIO port P1

Register overview

Register	Offset	Description
TASKS_ACQUIRE	0x014	Acquire SPI semaphore
TASKS_RELEASE	0x018	Release SPI semaphore, enabling the SPI slave to acquire it
TASKS_DMA.RX.ENABLEMATCH[n]	0x030	Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.RX.DISABLEMATCH[n]	0x040	Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
SUBSCRIBE_ACQUIRE	0x094	Subscribe configuration for task ACQUIRE
SUBSCRIBE_RELEASE	0x098	Subscribe configuration for task RELEASE
SUBSCRIBE_DMA.RX.ENABLEMATCH[n]	0x0B0	Subscribe configuration for task ENABLEMATCH[n]
SUBSCRIBE_DMA.RX.DISABLEMATCH[n]	0x0C0	Subscribe configuration for task DISABLEMATCH[n]
EVENTS_END	0x104	Granted transaction completed
EVENTS_ACQUIRED	0x118	Semaphore acquired
EVENTS_DMA.RX.END	0x14C	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.RX.READY	0x150	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.RX.BUSERROR	0x154	An error occured during the bus transfer.
EVENTS_DMA.RX.MATCH[n]	0x158	Pattern match is detected on the DMA data bus.
EVENTS_DMA.TX.END	0x168	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.TX.READY	0x16C	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.TX.BUSERROR	0x170	An error occured during the bus transfer.
PUBLISH_END	0x184	Publish configuration for event END
PUBLISH_ACQUIRED	0x198	Publish configuration for event ACQUIRED
PUBLISH_DMA.RX.END	0x1CC	Publish configuration for event END
PUBLISH_DMA.RX.READY	0x1D0	Publish configuration for event READY
PUBLISH_DMA.RX.BUSERROR	0x1D4	Publish configuration for event BUSERROR
PUBLISH_DMA.RX.MATCH[n]	0x1D8	Publish configuration for event MATCH[n]
PUBLISH_DMA.TX.END	0x1E8	Publish configuration for event END
PUBLISH_DMA.TX.READY	0x1EC	Publish configuration for event READY
PUBLISH_DMA.TX.BUSERROR	0x1F0	Publish configuration for event BUSERROR
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
CONFIG	0x554	Configuration register
DEF	0x55C	Default character. Character clocked out in case of an ignored transaction.
ORC	0x5C0	Over-read character
PSEL.SCK	0x600	Pin select for SCK
PSEL.MISO	0x604	Pin select for SDO signal
PSEL.MOSI	0x608	Pin select for SDI signal
PSEL.CSN	0x610	Pin select for CSN signal
DMA.RX.PTR	0x704	RAM buffer start address
DMA.RX.MAXCNT	0x708	Maximum number of bytes in channel buffer

Register	Offset	Description
DMA.RX.AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.RX.TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.RX.BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.
DMA.RX.MATCH.CONFIG	0x724	Configure individual match events
DMA.RX.MATCH.CANDIDATE[n]	0x728	The data to look for - any match will trigger the MATCH[n] event, if enabled.
DMA.TX.PTR	0x73C	RAM buffer start address
DMA.TX.MAXCNT	0x740	Maximum number of bytes in channel buffer
DMA.TX.AMOUNT	0x744	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.TX.TERMINATEONBUSERROR	0x754	Terminate the transaction if a BUSERROR event is detected.
DMA.TX.BUSERRORADDRESS	0x758	Address of transaction that generated the last BUSERROR event.

8.15.7.1 TASKS_ACQUIRE

Address offset: 0x014

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

8.15.7.2 TASKS_RELEASE

Address offset: 0x018

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

8.15.7.3 TASKS_DMA

Peripheral tasks.

8.15.7.3.1 TASKS_DMA.RX

Peripheral tasks.

8.15.7.3.1.1 TASKS_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x030 + (n × 0x4)

Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG 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	ENABLEMATCH			Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.15.7.3.1.2 TASKS_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG 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	DISABLEMATCH			Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.15.7.4 SUBSCRIBE_ACQUIRE

Address offset: 0x094

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task ACQUIRE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.15.7.5 SUBSCRIBE_RELEASE

Address offset: 0x098

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RELEASE will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.15.7.6 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.15.7.6.1 SUBSCRIBE_DMA.RX

Subscribe configuration for tasks

8.15.7.6.1.1 SUBSCRIBE_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task ENABLEMATCH[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		[0..255]	DPPI channel that task ENABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.15.7.6.1.2 SUBSCRIBE_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task DISABLEMATCH[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		[0..255]	DPPI channel that task DISABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.15.7.7 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

8.15.7.8 EVENTS_ACQUIRED

Address offset: 0x118

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 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

8.15.7.9 EVENTS_DMA

Peripheral events.

8.15.7.9.1 EVENTS_DMA.RX

Peripheral events.

8.15.7.9.1.1 EVENTS_DMA.RX.END

Address offset: 0x14C

Generated after all MAXCNT bytes have been transferred

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated after all MAXCNT bytes have been transferred
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.15.7.9.1.2 EVENTS_DMA.RX.READY

Address offset: 0x150

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.15.7.9.1.3 EVENTS_DMA.RX.BUSERROR

Address offset: 0x154

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR																																	
			NotGenerated	0																															
			Generated	1																															

A	RW	BUSERROR	An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
		NotGenerated	0 Event not generated
		Generated	1 Event generated

8.15.7.9.1.4 EVENTS_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x158 + (n × 0x4)

Pattern match is detected on the DMA data bus.

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

A	RW	MATCH	Pattern match is detected on the DMA data bus.
		NotGenerated	0 Event not generated
		Generated	1 Event generated

8.15.7.9.2 EVENTS_DMA.TX

Peripheral events.

8.15.7.9.2.1 EVENTS_DMA.TX.END

Address offset: 0x168

Generated after all MAXCNT bytes have been transferred

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

A	RW	END	Generated after all MAXCNT bytes have been transferred
		NotGenerated	0 Event not generated
		Generated	1 Event generated

8.15.7.9.2.2 EVENTS_DMA.TX.READY

Address offset: 0x16C

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.15.7.9.2.3 EVENTS_DMA.TX.BUSERROR

Address offset: 0x170

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.15.7.10 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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.11 PUBLISH_ACQUIRED

Address offset: 0x198

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		[0..255]	DPPI channel that event ACQUIRED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12 PUBLISH_DMA

Publish configuration for events

8.15.7.12.1 PUBLISH_DMA.RX

Publish configuration for events

8.15.7.12.1.1 PUBLISH_DMA.RX.END

Address offset: 0x1CC

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.1.2 PUBLISH_DMA.RX.READY

Address offset: 0x1D0

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.1.3 PUBLISH_DMA.RX.BUSERROR

Address offset: 0x1D4

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.1.4 PUBLISH_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x1D8 + (n × 0x4)

Publish configuration for event MATCH[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		[0..255]	DPPI channel that event MATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.2 PUBLISH_DMA.TX

Publish configuration for events

8.15.7.12.2.1 PUBLISH_DMA.TX.END

Address offset: 0x1E8

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.2.2 PUBLISH_DMA.TX.READY

Address offset: 0x1EC

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.12.2.3 PUBLISH_DMA.TX.BUSERROR

Address offset: 0x1F0

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.15.7.13 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				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	END_ACQUIRE			Shortcut between event END and task ACQUIRE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B-E	RW	DMA_RX_MATCH[i]_DMA_RX_ENABLEMA +1)%4] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.ENABLEMATCH[(i+1)%4] Allows daisy-chaining match events.
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F-I	RW	DMA_RX_MATCH[i]_DMA_RX_DISABLEMATCH[i] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.DISABLEMATCH[n]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.15.7.14 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				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 W1S	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	ACQUIRED			Write '1' to enable interrupt for event ACQUIRED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	DMARXEND			Write '1' to enable interrupt for event DMARXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	DMARXREADY			Write '1' to enable interrupt for event DMARXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	DMARXBUSERROR			Write '1' to enable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F-I	RW W1S	DMARXMATCH[i] (i=0..3)			Write '1' to enable interrupt for event DMARXMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1S	DMATXEND			Write '1' to enable interrupt for event DMATXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1S	DMATXREADY			Write '1' to enable interrupt for event DMATXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1S	DMATXBUSERROR			Write '1' to enable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.15.7.15 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				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 W1C	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	ACQUIRED			Write '1' to disable interrupt for event ACQUIRED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	DMARXEND			Write '1' to disable interrupt for event DMARXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	DMARXREADY			Write '1' to disable interrupt for event DMARXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	DMARXBUSERROR			Write '1' to disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F-I	RW W1C	DMARXMATCH[i] (i=0..3)			Write '1' to disable interrupt for event DMARXMATCH[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1C	DMATXEND			Write '1' to disable interrupt for event DMATXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW W1C	DMATXREADY			Write '1' to disable interrupt for event DMATXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	DMATXBUSERROR			Write '1' to disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.

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

8.15.7.16 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

8.15.7.17 STATUS

Address offset: 0x440

Status from last transaction

Individual bits are cleared by writing a '1' to the bits that shall be cleared.

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

8.15.7.18 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

8.15.7.19 CONFIG

Address offset: 0x554

Configuration register

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

8.15.7.20 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.

8.15.7.21 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.

8.15.7.22 PSEL.SCK

Address offset: 0x600

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.15.7.23 PSEL.MISO

Address offset: 0x604

Pin select for SDO 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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.15.7.24 PSEL.MOSI

Address offset: 0x608

Pin select for SDI 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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.15.7.25 PSEL.CSN

Address offset: 0x610

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.15.7.26 DMA.RX.PTR

Address offset: 0x704

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.15.7.27 DMA.RX.MAXCNT

Address offset: 0x708

Maximum number of bytes in channel 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 channel buffer

8.15.7.28 DMA.RX.AMOUNT

Address offset: 0x70C

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.15.7.29 DMA.RX.TERMINATEONBUSERROR

Address offset: 0x71C

Terminate the transaction if a BUSERROR event is 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	ENABLE																																	
				Disabled				0																		Disable									
				Enabled				1																		Enable									

8.15.7.30 DMA.RX.BUSERRORADDRESS

Address offset: 0x720

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	ADDRESS																																	

8.15.7.31 DMA.RX.MATCH

Registers to control the behavior of the pattern matcher engine

8.15.7.31.1 DMA.RX.MATCH.CONFIG

Address offset: 0x724

Configure individual match events

Bit number				31	30	29	28	27	26	25	24	23	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-D	RW	ENABLE[i] (i=0..3)																								Enable match filter i									
				Disabled				0																		Match filter disabled									
				Enabled				1																		Match filter enabled									
E-H	RW	ONESHOT[i] (i=0..3)																								Configure match filter i as one-shot or sticky One-shot match filters can be used together with shortcuts to check for continuous data sequences by disabling the filter if the next data is not a match. **Note:** The presence of these shorts depends on the configuration of the peripheral integrating this EasyDMA.									
				Continuous				0																		Match filter stays enabled until disabled by task									
				Oneshot				1																		Match filter stays enabled until next data word is received									

8.15.7.31.2 DMA.RX.MATCH.CANDIDATE[n] (n=0..3)

Address offset: 0x728 + (n × 0x4)

The data to look for - any match will trigger the MATCH[n] event, if enabled.

Note: This register can be updated while a transfer is in progress, but the new value will not take effect until a match has been found or the transfer is done. That makes it possible to write a new set of match words which will be searched for immediately after the event triggers.

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

8.15.7.32 DMA.TX.PTR

Address offset: 0x73C

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.15.7.33 DMA.TX.MAXCNT

Address offset: 0x740

Maximum number of bytes in channel 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 channel buffer

8.15.7.34 DMA.TX.AMOUNT

Address offset: 0x744

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.15.7.35 DMA.TX.TERMINATEONBUSERROR

Address offset: 0x754

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.15.7.36 DMA.TX.BUSERRORADDRESS

Address offset: 0x758

Address of transaction that generated the last BUSERROR 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 A A A A A A A A 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	ADDRESS			

8.16 TEMP — Temperature sensor

The temperature sensor peripheral (TEMP) 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 degrees
- TEMP analog electronics power down after temperature measurement is completed

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 54 for more information.

8.16.1 Operation

TEMP is started by triggering the START task.

When the temperature measurement is finished, a DATARDY event will be generated and the measurement result can be read from the TEMP register.

When the temperature measurement is finished, 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.

8.16.2 Registers

Instances

Instance	Domain	Base address	Description
TEMP	GLOBAL	0x400D7000	Temperature sensor TEMP

Register overview

Register	Offset	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 1st piece wise linear function
A1	0x524	Slope of 2nd piece wise linear function
A2	0x528	Slope of 3rd piece wise linear function
A3	0x52C	Slope of 4th piece wise linear function
A4	0x530	Slope of 5th piece wise linear function
A5	0x534	Slope of 6th piece wise linear function
A6	0x538	Slope of 7th piece wise linear function
B0	0x540	y-intercept of 1st piece wise linear function
B1	0x544	y-intercept of 2nd piece wise linear function
B2	0x548	y-intercept of 3rd piece wise linear function
B3	0x54C	y-intercept of 4th piece wise linear function
B4	0x550	y-intercept of 5th piece wise linear function
B5	0x554	y-intercept of 6th piece wise linear function
B6	0x558	y-intercept of 7th piece wise linear function
T0	0x560	End point of 1st piece wise linear function
T1	0x564	End point of 2nd piece wise linear function
T2	0x568	End point of 3rd piece wise linear function
T3	0x56C	End point of 4th piece wise linear function
T4	0x570	End point of 5th piece wise linear function
T5	0x574	End point of 6th piece wise linear function

8.16.2.1 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

8.16.2.2 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

8.16.2.3 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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.16.2.4 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.16.2.5 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

8.16.2.6 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		[0..255]	DPPI channel that event DATARDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.16.2.7 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
	W1S				
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.16.2.8 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
	W1C				
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.16.2.9 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 11 bits sign extended to 32 bits, with 2 LSBs as fractional bits. Decision point: DATARDY											

8.16.2.10 A0

Address offset: 0x520

Slope of 1st piece wise linear function

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x0000038C					**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**	**0**	**0**
ID	R/W	Field		Value ID					Value																Description											
A	RW	A0																							Slope of 1st piece wise linear function										

8.16.2.11 A1

Address offset: 0x524

Slope of 2nd piece wise linear function

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x000003B3					**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
ID	R/W	Field		Value ID					Value																Description											
A	RW	A1																							Slope of 2nd piece wise linear function										

8.16.2.12 A2

Address offset: 0x528

Slope of 3rd piece wise linear function

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x000003FA					**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**	**0**	**1**	**0**
ID	R/W	Field		Value ID					Value																Description											
A	RW	A2																							Slope of 3rd piece wise linear function										

8.16.2.13 A3

Address offset: 0x52C

Slope of 4th piece wise linear function

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x00000451					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	A3			Slope of 4th piece wise linear function

8.16.2.14 A4

Address offset: 0x530

Slope of 5th piece wise linear function

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x000004AA					**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 0 1 0 1 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	A4			Slope of 5th piece wise linear function

8.16.2.15 A5

Address offset: 0x534

Slope of 6th piece wise linear function

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x00000539					**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 0 0 1**
ID	R/W	Field	Value ID	Value	Description
A	RW	A5			Slope of 6th piece wise linear function

8.16.2.16 A6

Address offset: 0x538

Slope of 7th piece wise linear function

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A A A A A A A A A A A A A A A A
Reset 0x00000578					**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 1 1 1 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	A6			Slope of 7th piece wise linear function

8.16.2.17 B0

Address offset: 0x540

y-intercept of 1st piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000037	**0 0 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 1 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	B0			y-intercept of 1st piece wise linear function

8.16.2.18 B1

Address offset: 0x544

y-intercept of 2nd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x00000011	**0 0 0 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 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	B1			y-intercept of 2nd piece wise linear function

8.16.2.19 B2

Address offset: 0x548

y-intercept of 3rd piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 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	RW	B2			y-intercept of 3rd piece wise linear function

8.16.2.20 B3

Address offset: 0x54C

y-intercept of 4th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x0000002B	**0 0 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 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	B3			y-intercept of 4th piece wise linear function

8.16.2.21 B4

Address offset: 0x550

y-intercept of 5th piece wise linear function

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A
Reset 0x0000008F	**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 1 1 1 1**

ID	R/W	Field	Value ID	Value	Description
A	RW	B4			y-intercept of 5th piece wise linear function

8.16.2.22 B5

Address offset: 0x554

y-intercept of 6th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																					A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x0000015D	**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**	**1**	**1**	**0**	**1**

ID	R/W	Field	Value ID	Value	Description
A	RW	B5			y-intercept of 6th piece wise linear function

8.16.2.23 B6

Address offset: 0x558

y-intercept of 7th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																					A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x000001C0	**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**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	B6			y-intercept of 7th piece wise linear function

8.16.2.24 T0

Address offset: 0x560

End point of 1st piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x000000E5	**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**	**1**	**0**	**1**

ID	R/W	Field	Value ID	Value	Description
A	RW	T0			End point of 1st piece wise linear function

8.16.2.25 T1

Address offset: 0x564

End point of 2nd piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 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	T1			End point of 2nd piece wise linear function

8.16.2.26 T2

Address offset: 0x568

End point of 3rd piece wise linear function

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

8.16.2.27 T3

Address offset: 0x56C

End point of 4th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x0000002B	**0**	**0**	**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**	**1**

ID	R/W	Field	Value ID	Value	Description
A	RW	T3			End point of 4th piece wise linear function

8.16.2.28 T4

Address offset: 0x570

End point of 5th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000041	**0**	**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**	**1**

ID	R/W	Field	Value ID	Value	Description
A	RW	T4			End point of 5th piece wise linear function

8.16.2.29 T5

Address offset: 0x574

End point of 6th piece wise linear function

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A
Reset 0x00000050	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**

ID	R/W	Field	Value ID	Value	Description
A	RW	T5			End point of 6th piece wise linear function

8.17 TIMER — Timer/counter

The TIMER peripheral is a general purpose timer allowing time intervals to be defined by user input.

The main features of TIMER are:

- Two modes of operation: Timer mode and Counter mode
- Multiple capture/compare registers
- Compare event for every capture/compare registers
- 4-bit (1/2X) prescaler
- Configurable number of bits used by the TIMER: 8, 16, 24 or 32 bits
- TIMER runs on the high-frequency clock source (HFCLK)

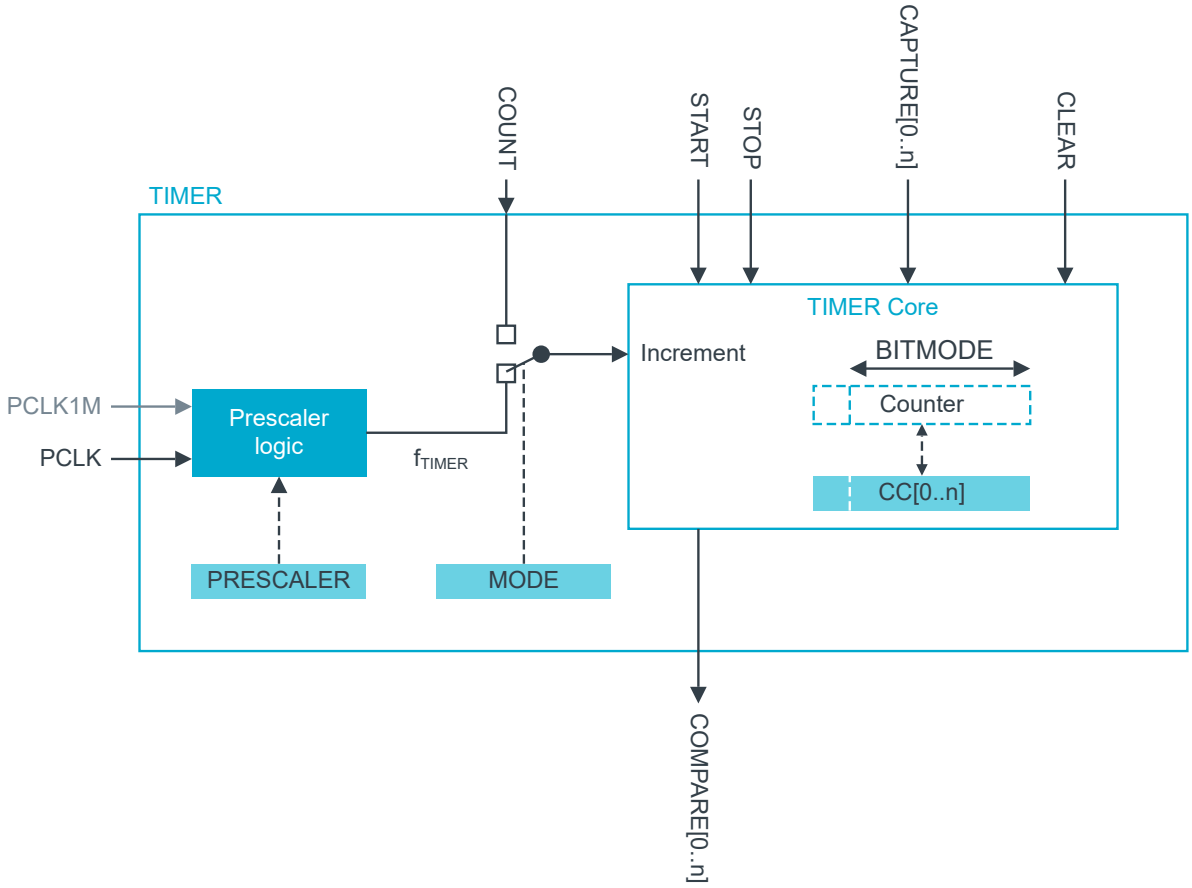


Figure 92: Block schematic for timer/counter

TIMER runs on the high-frequency clock source (HFCLK) and includes a four-bit (1/2X) prescaler that can divide the timer input clock (PCLK) from the HFCLK controller. The TIMER base frequency is always given as PCLK 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 413. The timer frequency is derived from PCLK as shown in the following example, using the values specified in the PRESCALER register.

$$f_{TIMER} = PCLK / (2^{PRESCALER})$$

For timers using PCLK16M as PCLK, when $f_{TIMER} \leq 1$ MHz, TIMER uses PCLK1M instead of PCLK for reduced power consumption. Clock source selection between PCLK and PCLK1M is automatic according to the TIMER base frequency set by the prescaler.

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 420.

PRESCALER on page 420 and BITMODE on page 420 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 413.

8.17.1 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.

8.17.2 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 420 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].

8.17.3 Task delays

After TIMER is started, the CLEAR, COUNT, and STOP tasks are guaranteed to take effect within one clock cycle of the PCLK.

8.17.4 Task priority

If the START task and the STOP task are triggered at the same time, meaning within the same period of PCLK, 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 PCLK, the CAPTURE tasks are prioritized. This means that the CC registers will capture the counter value before the CLEAR tasks are triggered.

8.17.5 Registers

Instances

Instance	Domain	Base address	Description
TIMER00	GLOBAL	0x40055000	Timer TIMER00
TIMER10	GLOBAL	0x40085000	Timer TIMER10
TIMER20	GLOBAL	0x400CA000	Timer TIMER20

Configuration

Instance	Domain	Configuration
TIMER00	GLOBAL	Peripheral clock frequency (PCLK) is 128 MHz The system is able to configure the TIMER peripheral input clock frequency (PCLK) before it reaches TIMER, and calculations of PRESCALER value must take the actual PCLK frequency into account 6 capture compare channels implemented
TIMER10	GLOBAL	Peripheral clock frequency (PCLK) is 32 MHz 8 capture compare channels implemented
TIMER20	GLOBAL	Peripheral clock frequency (PCLK) is 16 MHz 6 capture compare channels implemented

Register overview

Register	Offset	Description
TASKS_START	0x000	Start Timer
TASKS_STOP	0x004	Stop Timer
TASKS_COUNT	0x008	Increment Timer (Counter mode only)
TASKS_CLEAR	0x00C	Clear time
TASKS_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_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

8.17.5.1 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

8.17.5.2 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

8.17.5.3 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

8.17.5.4 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

8.17.5.5 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

8.17.5.6 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.17.5.7 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.17.5.8 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task COUNT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.17.5.9 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CLEAR will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.17.5.10 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task CAPTURE[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.17.5.11 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

8.17.5.12 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		[0..255]	DPPI channel that event COMPARE[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.17.5.13 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																							

8.17.5.14 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																							

8.17.5.15 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]																							
	W1S																																		
			Set	1								Enable																							
			Disabled	0								Read: Disabled																							
			Enabled	1								Read: Enabled																							

8.17.5.16 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]
	W1C				
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.17.5.17 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
					This enumerator is deprecated.
			LowPowerCounter	2	Select Low Power Counter mode

8.17.5.18 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

8.17.5.19 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

8.17.5.20 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																															
A	RW	CC																																	

Capture/Compare value

Only the number of bits indicated by BITMODE will be used by the TIMER.

8.17.5.21 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																															
A	RW	ONESHOTEN																																	
			Disable	0																															
			Enable	1																															

Configures the corresponding compare-channel for one-shot operation

Disable one-shot operation

Compare event is generated every time the Counter matches CC[n]

Enable one-shot operation

Compare event is generated the first time the Counter matches CC[n] after CC[n] has been written

8.18 TWIM — I²C compatible two-wire interface controller with EasyDMA

The TWI controller peripheral (TWIM) with EasyDMA provides a half duplex, two-wire synchronous serial communication interface which supports multiple targets in the same bus.

The main features of TWIM are the following:

- I²C compatible for 100 kbps and 400 kbps
- 1000 kbps bit rate support for selected pull-up resistor/bus capacitance combinations
- Supported baud rates:
  - 100 kbps
  - 400 kbps
  - 1000 kbps
- EasyDMA direct transfer to and from RAM
- Individual selection of I/O pins
- Support for clock stretching
- Transmissions can be suspended and resumed

The two-wire interface can communicate with a bidirectional wired-AND bus with two lines (SCL, SDA). The interface enables interconnecting up to 127 individually addressable devices. TWIM is not compatible with CBUS.

Selecting GPIO pins individually ensures flexibility in device pinout and efficient use of board space and signal routing.

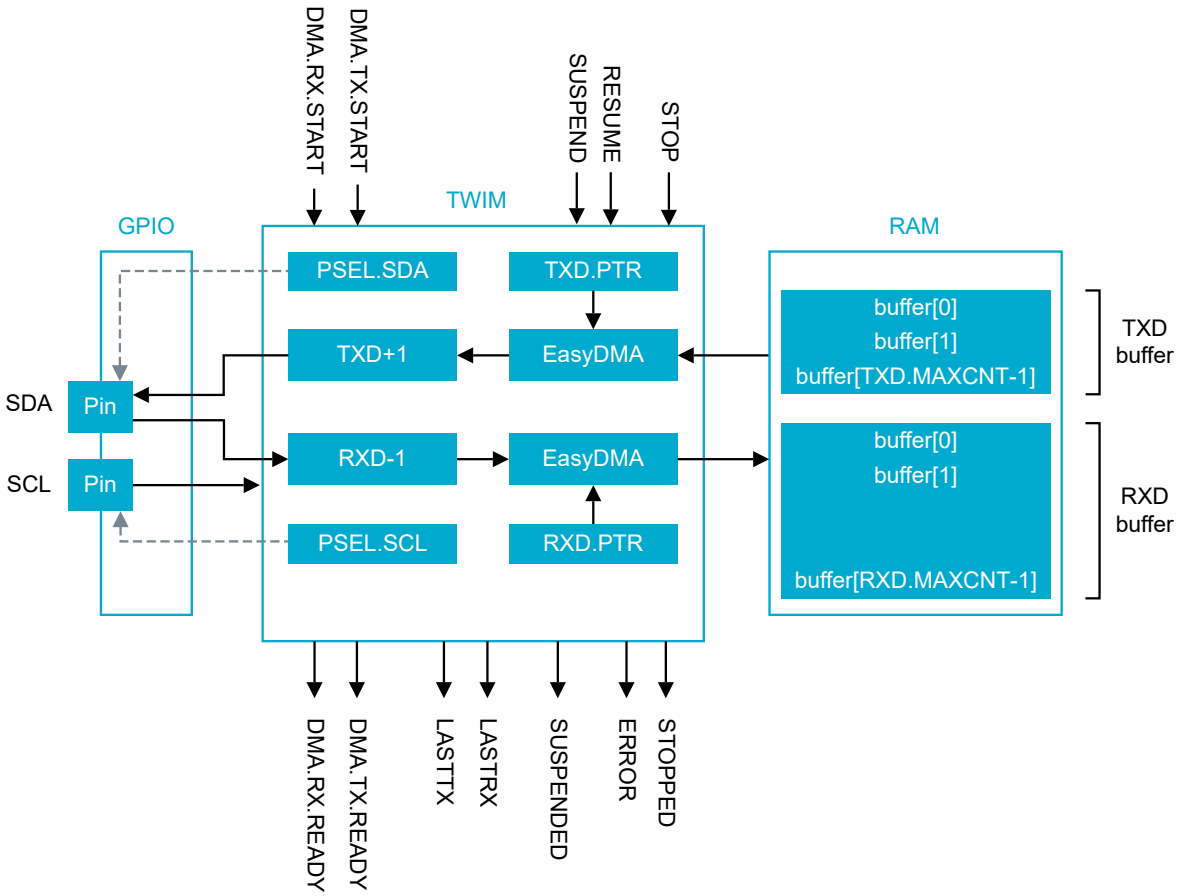


Figure 93: TWIM with EasyDMA

A typical TWIM setup consists of one controller and one or more targets, as illustrated in the following figure. TWIM can only operate as a single controller on the TWI bus. A bus configuration with multiple controllers is not supported.

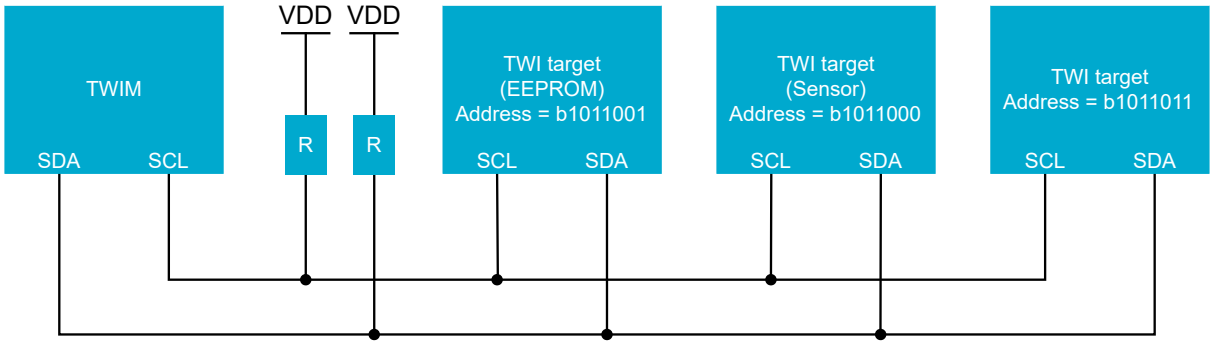


Figure 94: A typical TWIM setup with one controller and three targets

TWIM supports clock stretching performed by the targets.

8.18.1 TWIM operation

TWIM is started by triggering the DMA.TX.START or DMA.RX.START tasks, and stopped by triggering the STOP task. After a STOP task, TWIM generates a STOPPED event when it has stopped.

After TWIM starts, the DMA.TX.START and DMA.RX.START tasks must not be triggered again until TWIM has issued a LASTRX, LASTTX, or STOPPED event.

TWIM can be suspended using the SUSPEND task, which is useful when using TWIM in a low priority interrupt context. When TWIM enters the SUSPEND state, it will automatically issue a SUSPENDED event while performing a continuous clock stretching. This continues until a RESUME task is received. TWIM cannot be stopped while it is suspended. The STOP task must be issued after TWIM resumes operation.

Note: Any ongoing byte transfer is allowed to complete before 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 target, TWIM generates an ERROR event.

8.18.2 Shared resources

The TWIM peripheral shares registers and other resources with peripherals that have the same ID as TWIM. Therefore, all peripherals that have the same ID as TWIM must be disabled before TWIM can be configured and used.

Disabling shared peripherals will not reset any of the registers that are shared with TWIM. Configure all relevant TWIM registers to ensure they operate correctly.

See the Instantiation table in Instantiation on page 141 for details on peripherals and their IDs.

8.18.3 EasyDMA

EasyDMA is implemented by TWIM in order to access RAM without the CPU.

TWIM implements the following EasyDMA channels.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 46: TWIM EasyDMA channels

The RXD.PTR, TXD.PTR, RXD.MAXCNT, and TXD.MAXCNT registers are double-buffered. They are ready for the next transmission immediately after receiving the EVENTS_DMA.RX.READY or EVENTS_DMA.TX.READY event.

The STOPPED event indicates that EasyDMA is finished accessing the buffer in RAM.

See EasyDMA on page 19 for more detailed information.

8.18.4 TWIM write sequence

A TWIM write sequence is started by triggering the DMA.TX.START task. After the DMA.TX.START task has been triggered, TWIM generates a start condition on the TWI bus. This is followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE = 0, READ = 1).

The target device address the controller wants to write to must match the clocked address. The READ/WRITE bit is followed by an ACK/NACK bit (ACK = 0 or NACK = 1) generated by the target.

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 is followed by an ACK/NACK bit clocked in from the target.

A typical TWIM write sequence including clock stretching performed by TWIM following a SUSPEND task is shown in the following figure.

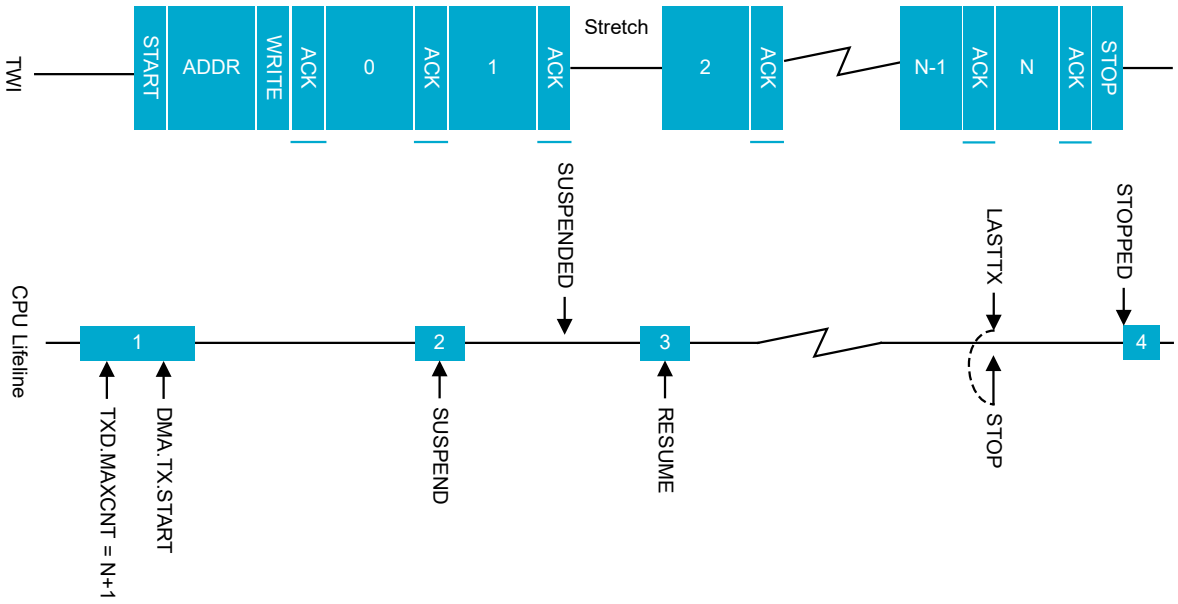


Figure 95: TWIM writing data to a target

A SUSPENDED event indicates that the SUSPEND task has taken effect.

TWIM will generate a LASTTX event when it starts to transmit the last byte.

TWIM is stopped by triggering the STOP task. To stop TWIM as fast as possible, trigger the task during the transmission of the last byte. The shortcut between LASTTX and STOP can also be used to accomplish this.

TWIM does not stop on its own when the entire RAM buffer has been sent or when an error occurs. The STOP task must be issued, either through the local or PPI shortcut, or in software as part of the error handler.

8.18.5 TWIM read sequence

A TWIM read sequence is started by triggering the DMA.RX.START task. After the DMA.RX.START task has been triggered, TWIM generates a start condition on the TWI bus. This is 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 target device that the controller wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK = 0 or NACK = 1) generated by the target.

After sending the ACK bit, the TWI target sends data to the controller using the clock generated by TWIM.

Data received will be stored in RAM at the address specified in the RXD.PTR register. TWIM will generate an ACK before the last byte is received from the target. TWIM generates a NACK after the last byte received to indicate that the read sequence will stop.

A typical TWIM read sequence is illustrated in the following figure, including clock stretching performed by TWIM following a SUSPEND task.

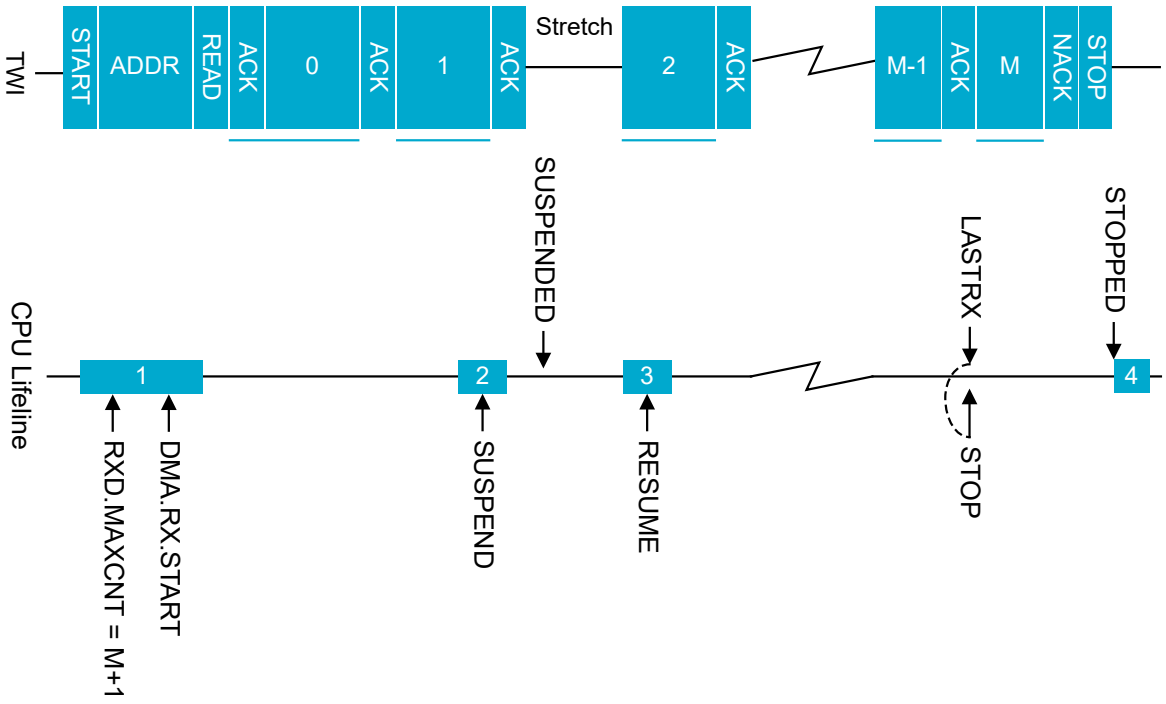


Figure 96: TWIM reading data from a target

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 begins transmission. 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.

TWIM does not stop on its own when the RAM buffer is full or when an error occurs. The STOP task must be issued, either through a local or PPI shortcut, or in software as part of the error handler.

TWIM cannot be stopped while suspended. The STOP task must be issued after TWIM has been resumed.

8.18.6 TWIM repeated start sequence

A typical repeated start sequence is when TWIM writes two bytes to the target followed by reading four bytes from the target. This example uses shortcuts to perform a simple repeated start sequence, with 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 a write.

The following figure shows an example of a repeated start sequence where TWIM writes two bytes followed by reading four bytes from the target.

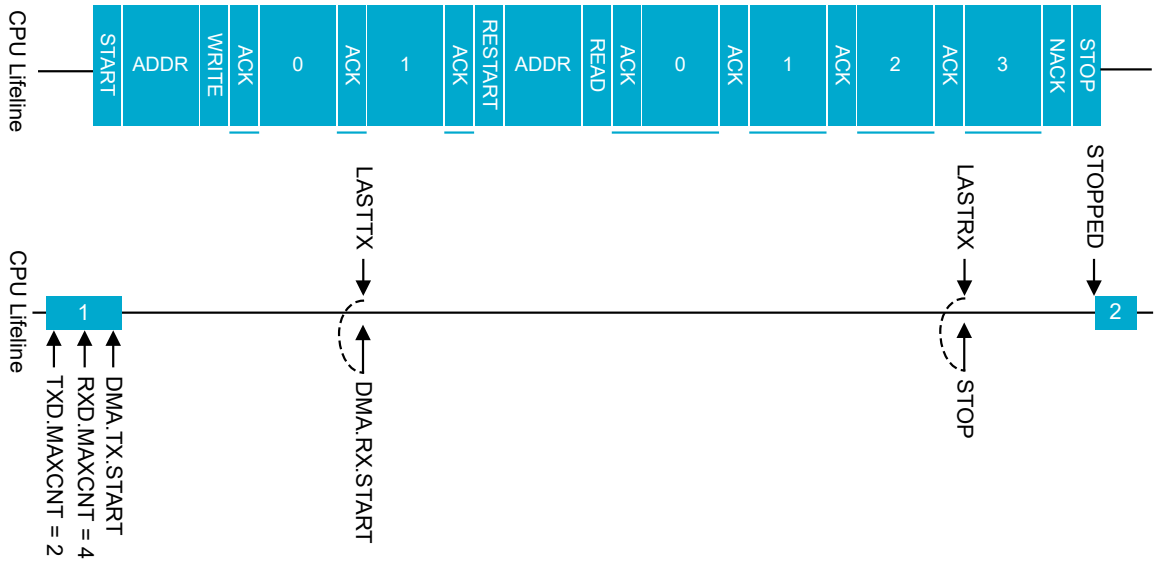


Figure 97: Controller repeated start sequence

If a more complex repeated start sequence is needed, and the TWI firmware drive is serviced in a low priority interrupt, use the SUSPEND task and SUSPENDED event to ensure 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.

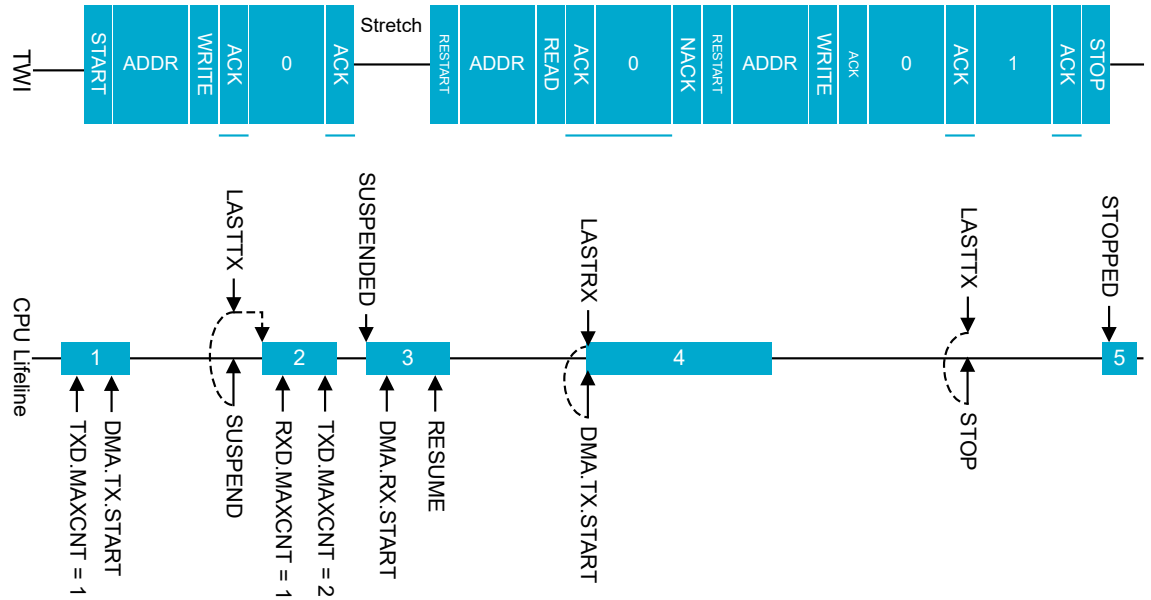


Figure 98: Double repeated start sequence

8.18.7 Low power

When the peripheral is not needed, stop and disable TWIM for lowest possible power consumption.

When the STOP task is sent, the software must wait until the STOPPED event is received before disabling the peripheral through the ENABLE register. If the peripheral is already stopped, the STOP task is not needed.

8.18.8 TWIM pin configuration

The SCL and SDA signals are mapped to physical pins using the PSEL.SCL and PSEL.SDA registers.

These registers and their configurations are only used when TWIM is enabled, and retained while the device is in System ON mode. When the peripheral is disabled, the pins behave as regular GPIOs and are configured according to their respective OUT bit field and PIN_CNF[n] register. Configure registers PSEL.SCL and PSEL.SDA when TWIM is disabled.

Only one peripheral can be assigned to drive a GPIO pin at a time. If more than one peripheral is assigned to a GPIO pin, it could result in unpredictable behavior.

When TWIM is in System OFF mode or disabled, the pins using TWIM must be configured by the GPIO peripheral according to the following table to ensure correct pin behavior.

TWIM signal	TWIM pin	Drive strength	Direction	Output value
SCL	As specified in PSEL.SCL	S0D1	Input	Not applicable
SDA	As specified in PSEL.SDA	S0D1	Input	Not applicable

Table 47: GPIO configuration before enabling peripheral

8.18.9 Pull-up resistor

1000 kbps bit rate is supported when using H0D1 drive strength, 1 kΩ pull-up resistor, and maximum 50 pF bus capacitance. For other bit rates, see the following figure.

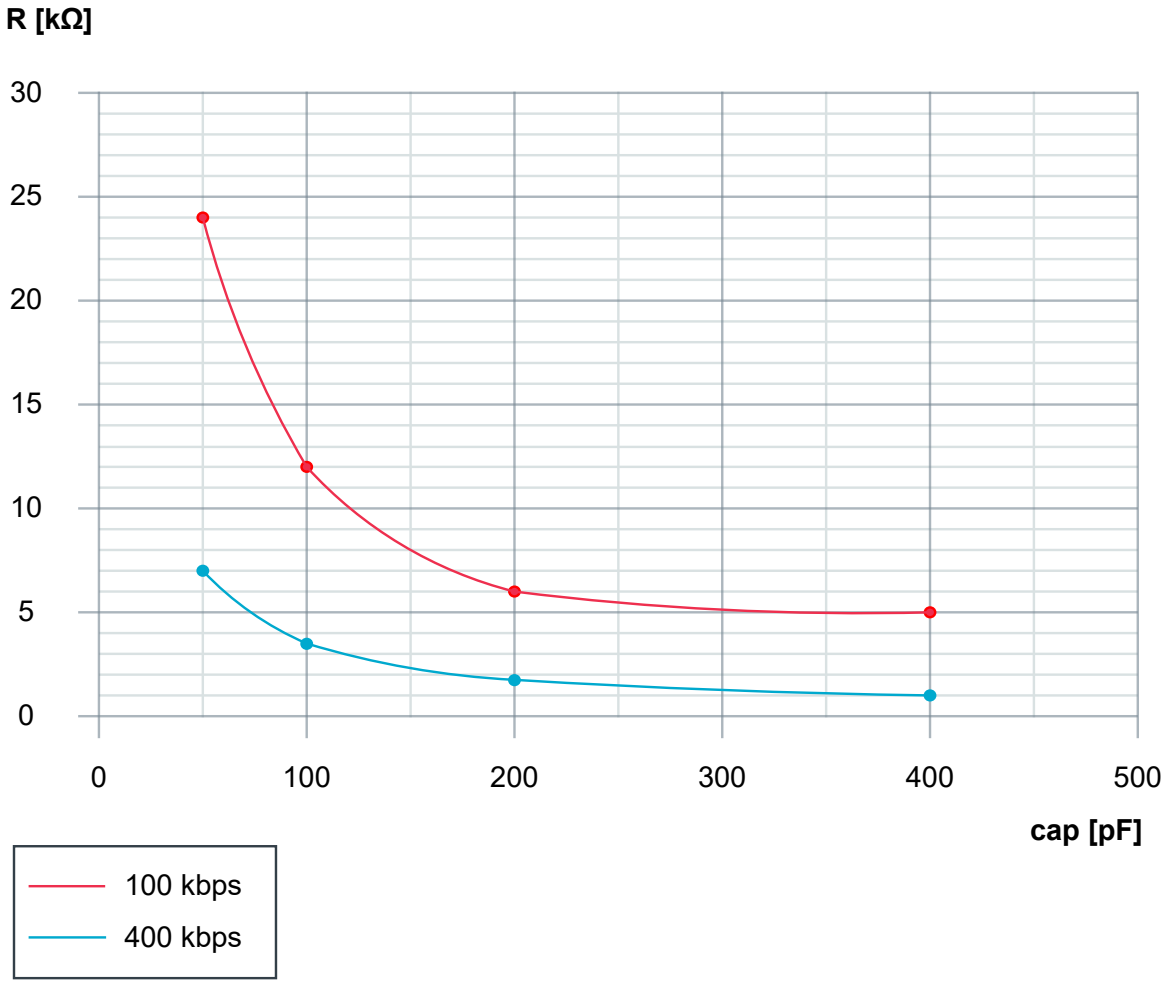


Figure 99: Recommended TWIM pull-up value vs. line capacitance

- The *I^2C bus specification* allows a maximum line capacitance of 400 pF.
- The value of internal pull-up resistor (R_{PU}) for nRF54LS05A/nRF54LS05B can be found in GPIO — General purpose input/output on page 182.

8.18.10 Registers

Instances

Instance	Domain	Base address	Description
TWIM20	GLOBAL	0x400C6000	Two-wire interface controller TWIM20
TWIM21	GLOBAL	0x400C7000	Two-wire interface controller TWIM21
TWIM22	GLOBAL	0x400C8000	Two-wire interface controller TWIM22

Configuration

Instance	Domain	Configuration
TWIM20	GLOBAL	Use GPIO port P1
TWIM21	GLOBAL	Use GPIO port P1
TWIM22	GLOBAL	Use GPIO port P1

Register overview

Register	Offset	Description
TASKS_STOP	0x004	Stop TWI transaction. Must be issued while the TWI master is not suspended.
TASKS_SUSPEND	0x00C	Suspend TWI transaction
TASKS_RESUME	0x010	Resume TWI transaction
TASKS_DMA.RX.START	0x028	Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
TASKS_DMA.RX.STOP	0x02C	Stops operation using easyDMA. This does not trigger an END event.
TASKS_DMA.RX.ENABLEMATCH[n]	0x030	Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.RX.DISABLEMATCH[n]	0x040	Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.TX.START	0x050	Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
TASKS_DMA.TX.STOP	0x054	Stops operation using easyDMA. This does not trigger an END event.
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x08C	Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x090	Subscribe configuration for task RESUME
SUBSCRIBE_DMA.RX.START	0x0A8	Subscribe configuration for task START
SUBSCRIBE_DMA.RX.STOP	0x0AC	Subscribe configuration for task STOP
SUBSCRIBE_DMA.RX.ENABLEMATCH[n]	0x0B0	Subscribe configuration for task ENABLEMATCH[n]
SUBSCRIBE_DMA.RX.DISABLEMATCH[n]	0x0C0	Subscribe configuration for task DISABLEMATCH[n]
SUBSCRIBE_DMA.TX.START	0x0D0	Subscribe configuration for task START
SUBSCRIBE_DMA.TX.STOP	0x0D4	Subscribe configuration for task STOP
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x114	TWI error
EVENTS_SUSPENDED	0x128	SUSPEND task has been issued, TWI traffic is now suspended.
EVENTS_LASTRX	0x134	Byte boundary, starting to receive the last byte
EVENTS_LASTTX	0x138	Byte boundary, starting to transmit the last byte
EVENTS_DMA.RX.END	0x14C	Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully completed.

Register	Offset	Description
EVENTS_DMA.RX.READY	0x150	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.RX.BUSERROR	0x154	An error occured during the bus transfer.
EVENTS_DMA.RX.MATCH[n]	0x158	Pattern match is detected on the DMA data bus.
EVENTS_DMA.TX.END	0x168	Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully completed.
EVENTS_DMA.TX.READY	0x16C	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.TX.BUSERROR	0x170	An error occured during the bus transfer.
PUBLISH_STOPPED	0x184	Publish configuration for event STOPPED
PUBLISH_ERROR	0x194	Publish configuration for event ERROR
PUBLISH_SUSPENDED	0x1A8	Publish configuration for event SUSPENDED
PUBLISH_LASTRX	0x1B4	Publish configuration for event LASTRX
PUBLISH_LASTTX	0x1B8	Publish configuration for event LASTTX
PUBLISH_DMA.RX.END	0x1CC	Publish configuration for event END
PUBLISH_DMA.RX.READY	0x1D0	Publish configuration for event READY
PUBLISH_DMA.RX.BUSERROR	0x1D4	Publish configuration for event BUSERROR
PUBLISH_DMA.RX.MATCH[n]	0x1D8	Publish configuration for event MATCH[n]
PUBLISH_DMA.TX.END	0x1E8	Publish configuration for event END
PUBLISH_DMA.TX.READY	0x1EC	Publish configuration for event READY
PUBLISH_DMA.TX.BUSERROR	0x1F0	Publish configuration for event BUSERROR
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
FREQUENCY	0x524	TWI frequency. Accuracy depends on the HFCLK source selected.
ADDRESS	0x588	Address used in the TWI transfer
PSEL.SCL	0x600	Pin select for SCL signal
PSEL.SDA	0x604	Pin select for SDA signal
DMA.RX.PTR	0x704	RAM buffer start address
DMA.RX.MAXCNT	0x708	Maximum number of bytes in channel buffer
DMA.RX.AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.RX.TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.RX.BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.
DMA.RX.MATCH.CONFIG	0x724	Configure individual match events
DMA.RX.MATCH.CANDIDATE[n]	0x728	The data to look for - any match will trigger the MATCH[n] event, if enabled.
DMA.TX.PTR	0x73C	RAM buffer start address
DMA.TX.MAXCNT	0x740	Maximum number of bytes in channel buffer
DMA.TX.AMOUNT	0x744	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.TX.TERMINATEONBUSERROR	0x754	Terminate the transaction if a BUSERROR event is detected.
DMA.TX.BUSERRORADDRESS	0x758	Address of transaction that generated the last BUSERROR event.

8.18.10.1 TASKS_STOP

Address offset: 0x004

Stop TWI transaction. Must be issued while the TWI master is not 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	W	TASKS_STOP			Stop TWI transaction. Must be issued while the TWI master is not suspended.
			Trigger	1	Trigger task

8.18.10.2 TASKS_SUSPEND

Address offset: 0x00C

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

8.18.10.3 TASKS_RESUME

Address offset: 0x010

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

8.18.10.4 TASKS_DMA

Peripheral tasks.

8.18.10.4.1 TASKS_DMA.RX

Peripheral tasks.

8.18.10.4.1.1 TASKS_DMA.RX.START

Address offset: 0x028

Starts operation using easyDMA to load the values. See peripheral description for operation using 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	W	START			Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
			Trigger	1	Trigger task

8.18.10.4.1.2 TASKS_DMA.RX.STOP

Address offset: 0x02C

Stops operation using easyDMA. This does not trigger an END 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	STOP			Stops operation using easyDMA. This does not trigger an END event.
			Trigger	1	Trigger task

8.18.10.4.1.3 TASKS_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x030 + (n × 0x4)

Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG 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	ENABLEMATCH			Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.18.10.4.1.4 TASKS_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG 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	DISABLEMATCH			Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.18.10.4.2 TASKS_DMA.TX

Peripheral tasks.

8.18.10.4.2.1 TASKS_DMA.TX.START

Address offset: 0x050

Starts operation using easyDMA to load the values. See peripheral description for operation using 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	W	START			Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
			Trigger	1	Trigger task

8.18.10.4.2.2 TASKS_DMA.TX.STOP

Address offset: 0x054

Stops operation using easyDMA. This does not trigger an END 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	STOP			Stops operation using easyDMA. This does not trigger an END event.
			Trigger	1	Trigger task

8.18.10.5 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.6 SUBSCRIBE_SUSPEND

Address offset: 0x08C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.7 SUBSCRIBE_RESUME

Address offset: 0x090

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.18.10.8.1 SUBSCRIBE_DMA.RX

Subscribe configuration for tasks

8.18.10.8.1.1 SUBSCRIBE_DMA.RX.START

Address offset: 0x0A8

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8.1.2 SUBSCRIBE_DMA.RX.STOP

Address offset: 0x0AC

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8.1.3 SUBSCRIBE_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task ENABLEMATCH[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		[0..255]	DPPI channel that task ENABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8.1.4 SUBSCRIBE_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task DISABLEMATCH[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		[0..255]	DPPI channel that task DISABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8.2 SUBSCRIBE_DMA.TX

Subscribe configuration for tasks

8.18.10.8.2.1 SUBSCRIBE_DMA.TX.START

Address offset: 0x0D0

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.8.2.2 SUBSCRIBE_DMA.TX.STOP

Address offset: 0x0D4

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.18.10.9 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

8.18.10.10 EVENTS_ERROR

Address offset: 0x114

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

8.18.10.11 EVENTS_SUSPENDED

Address offset: 0x128

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

8.18.10.12 EVENTS_LASTRX

Address offset: 0x134

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 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

8.18.10.13 EVENTS_LASTTX

Address offset: 0x138

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 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

8.18.10.14 EVENTS_DMA

Peripheral events.

8.18.10.14.1 EVENTS_DMA.RX

Peripheral events.

8.18.10.14.1.1 EVENTS_DMA.RX.END

Address offset: 0x14C

Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully 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	END			Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully completed.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.1.2 EVENTS_DMA.RX.READY

Address offset: 0x150

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.1.3 EVENTS_DMA.RX.BUSERROR

Address offset: 0x154

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.1.4 EVENTS_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x158 + (n × 0x4)

Pattern match is detected on the DMA data bus.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MATCH			Pattern match is detected on the DMA data bus.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.2 EVENTS_DMA.TX

Peripheral events.

8.18.10.14.2.1 EVENTS_DMA.TX.END

Address offset: 0x168

Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully 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	END			Indicates that the transfer of MAXCNT bytes between memory and the peripheral has been fully completed.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.2.2 EVENTS_DMA.TX.READY

Address offset: 0x16C

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.14.2.3 EVENTS_DMA.TX.BUSERROR

Address offset: 0x170

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.18.10.15 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		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.16 PUBLISH_ERROR

Address offset: 0x194

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.17 PUBLISH_SUSPENDED

Address offset: 0x1A8

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event SUSPENDED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.18 PUBLISH_LASTRX

Address offset: 0x1B4

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		[0..255]	DPPI channel that event LASTRX will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.19 PUBLISH_LASTTX

Address offset: 0x1B8

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		[0..255]	DPPI channel that event LASTTX will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20 PUBLISH_DMA

Publish configuration for events

8.18.10.20.1 PUBLISH_DMA.RX

Publish configuration for events

8.18.10.20.1.1 PUBLISH_DMA.RX.END

Address offset: 0x1CC

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.1.2 PUBLISH_DMA.RX.READY

Address offset: 0x1D0

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.1.3 PUBLISH_DMA.RX.BUSERROR

Address offset: 0x1D4

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.1.4 PUBLISH_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x1D8 + (n × 0x4)

Publish configuration for event MATCH[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		[0..255]	DPPI channel that event MATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.2 PUBLISH_DMA.TX

Publish configuration for events

8.18.10.20.2.1 PUBLISH_DMA.TX.END

Address offset: 0x1E8

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.2.2 PUBLISH_DMA.TX.READY

Address offset: 0x1EC

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.20.2.3 PUBLISH_DMA.TX.BUSERROR

Address offset: 0x1F0

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.18.10.21 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				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	LASTTX_DMA_RX_START			Shortcut between event LASTTX and task DMA.RX.START
			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_DMA_TX_START			Shortcut between event LASTRX and task DMA.TX.START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LASTRX_STOP			Shortcut between event LASTRX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F-I	RW	DMA_RX_MATCH[i]_DMA_RX_ENABLEMA			Shortcut between event DMA.RX.MATCH[n] and task
		+1)%4] (i=0..3)			DMA.RX.ENABLEMATCH[(i+1)%4]
					Allows daisy-chaining match events.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																
				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	Disable shortcut
			Enabled	1	Enable shortcut
J-M	RW	DMA_RX_MATCH[i]_DMA_RX_DISABLEMATCH[i] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.DISABLEMATCH[n]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.18.10.22 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																																
				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	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
C	RW	SUSPENDED			Enable or disable interrupt for event SUSPENDED
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	LASTRX			Enable or disable interrupt for event LASTRX
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	LASTTX			Enable or disable interrupt for event LASTTX
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	DMARXEND			Enable or disable interrupt for event DMARXEND
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	DMARXREADY			Enable or disable interrupt for event DMARXREADY
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	DMARXBUSERROR			Enable or disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable
I-L	RW	DMARXMATCH[i] (i=0..3)			Enable or disable interrupt for event DMARXMATCH[i]
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	DMATXEND			Enable or disable interrupt for event DMATXEND
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	DMATXREADY			Enable or disable interrupt for event DMATXREADY

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				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	Disable
			Enabled	1	Enable
O	RW	DMATXBUSERROR			Enable or disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable

8.18.10.23 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				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 W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	SUSPENDED			Write '1' to enable interrupt for event SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	LASTRX			Write '1' to enable interrupt for event LASTRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	LASTTX			Write '1' to enable interrupt for event LASTTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	DMARXEND			Write '1' to enable interrupt for event DMARXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	DMARXREADY			Write '1' to enable interrupt for event DMARXREADY

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				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
H	RW W1S	DMARXBUSERROR			Write '1' to enable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I-L	RW W1S	DMARXMATCH[i] (i=0..3)			Write '1' to enable interrupt for event DMARXMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	DMATXEND			Write '1' to enable interrupt for event DMATXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1S	DMATXREADY			Write '1' to enable interrupt for event DMATXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1S	DMATXBUSERROR			Write '1' to enable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.18.10.24 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				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 W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	ERROR			Write '1' to disable interrupt for event ERROR
			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				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
C	RW W1C	SUSPENDED			Write '1' to disable interrupt for event SUSPENDED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	LASTRX			Write '1' to disable interrupt for event LASTRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	LASTTX			Write '1' to disable interrupt for event LASTTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	DMARXEND			Write '1' to disable interrupt for event DMARXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1C	DMARXREADY			Write '1' to disable interrupt for event DMARXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1C	DMARXBUSERROR			Write '1' to disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I-L	RW W1C	DMARXMATCH[i] (i=0..3)			Write '1' to disable interrupt for event DMARXMATCH[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	DMATXEND			Write '1' to disable interrupt for event DMATXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	DMATXREADY			Write '1' to disable interrupt for event DMATXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1C	DMATXBUSERROR			Write '1' to disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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				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
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.18.10.25 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 W1C	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 W1C	ANACK			NACK received after sending the address (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW W1C	DNACK			NACK received after sending a data byte (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred

8.18.10.26 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

8.18.10.27 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

8.18.10.28 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

8.18.10.29 PSEL.SCL

Address offset: 0x600

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.18.10.30 PSEL.SDA

Address offset: 0x604

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.18.10.31 DMA.RX.PTR

Address offset: 0x704

RAM buffer 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 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																															
A	RW	PTR																																	

RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.18.10.32 DMA.RX.MAXCNT

Address offset: 0x708

Maximum number of bytes in channel 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																															
A	RW	MAXCNT		[1..0xFFFF]																															

Maximum number of bytes in channel buffer

8.18.10.33 DMA.RX.AMOUNT

Address offset: 0x70C

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	R	AMOUNT		[1..0xFFFF]																															

Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

8.18.10.34 DMA.RX.TERMINATEONBUSERROR

Address offset: 0x71C

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.18.10.35 DMA.RX.BUSERRORADDRESS

Address offset: 0x720

Address of transaction that generated the last BUSERROR 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 A A A A A A A A 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	ADDRESS			

8.18.10.36 DMA.RX.MATCH

Registers to control the behavior of the pattern matcher engine

8.18.10.36.1 DMA.RX.MATCH.CONFIG

Address offset: 0x724

Configure individual match events

Bit number				31 30 29 28 27 26 25 24 23 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-D	RW	ENABLE[i] (i=0..3)			Enable match filter i
			Disabled	0	Match filter disabled
			Enabled	1	Match filter enabled
E-H	RW	ONESHOT[i] (i=0..3)			Configure match filter i as one-shot or continous One-shot operation will disable the filter on a match, while Continous operation will keep it enabled until explicitly disabled.
			Continuous	0	Match filter stays enabled until disabled by task
			Oneshot	1	Match filter stays enabled until next data word is received

8.18.10.36.2 DMA.RX.MATCH.CANDIDATE[n] (n=0..3)

Address offset: 0x728 + (n × 0x4)

The data to look for - any match will trigger the MATCH[n] event, if enabled.

Note: This register can be updated while a transfer is in progress, but the new value will not take effect until either the DMA is restarted or the match event is generated. That makes it possible to write a new set of match words which will be searched for immediately after the event triggers.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	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	DATA			Data to look for

8.18.10.37 DMA.TX.PTR

Address offset: 0x73C

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.18.10.38 DMA.TX.MAXCNT

Address offset: 0x740

Maximum number of bytes in channel 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 channel buffer

8.18.10.39 DMA.TX.AMOUNT

Address offset: 0x744

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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.

8.18.10.40 DMA.TX.TERMINATEONBUSERROR

Address offset: 0x754

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.18.10.41 DMA.TX.BUSERRORADDRESS

Address offset: 0x758

Address of transaction that generated the last BUSERROR 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 A A A A A A A A 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	ADDRESS			

8.19 TWIS — I²C compatible two-wire interface target with EasyDMA

The TWI target peripheral (TWIS) with EasyDMA provides a half duplex, two-wire synchronous serial communication interface.

The main features of TWIS are the following:

- I²C compatible
- Supports 100 kbps and 400 kbps bit rate
- EasyDMA direct transfer to and from RAM
- Individual selection of I/O pins
- Support for clock stretching

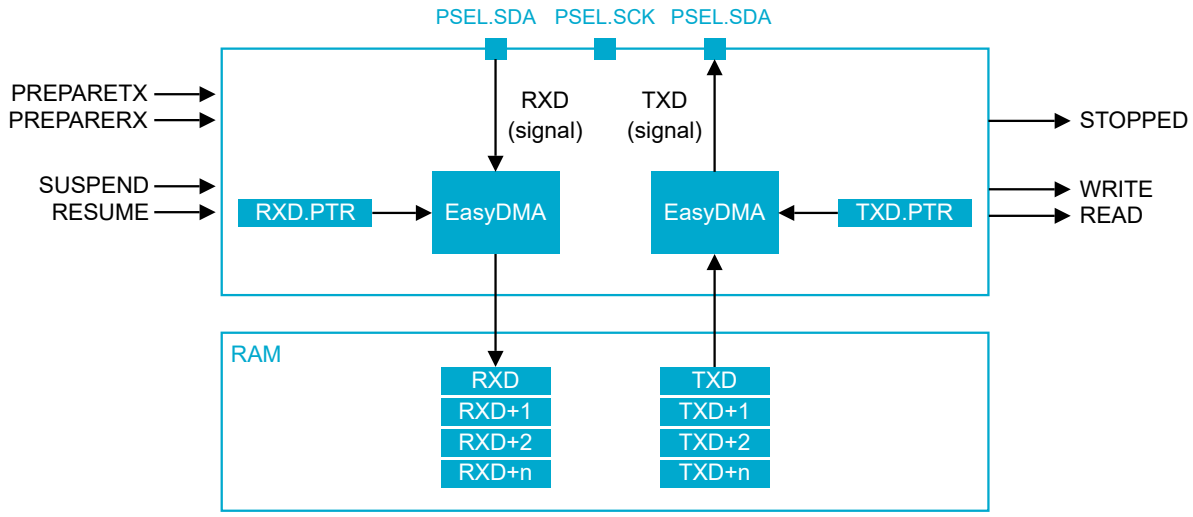


Figure 100: TWIS with EasyDMA

A typical TWI setup consists of one controller and one or more targets, as seen in the following figure. Only a single controller can be used on the TWI bus.

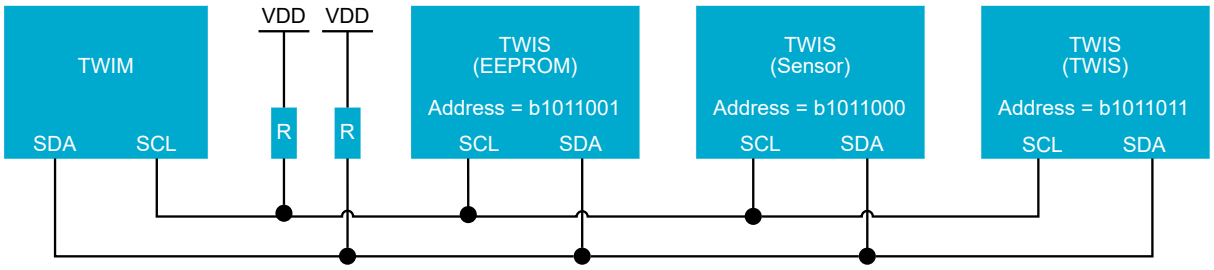


Figure 101: Typical TWI setup with one controller and three targets

8.19.1 State machine

The following figure shows the TWIS state machine.

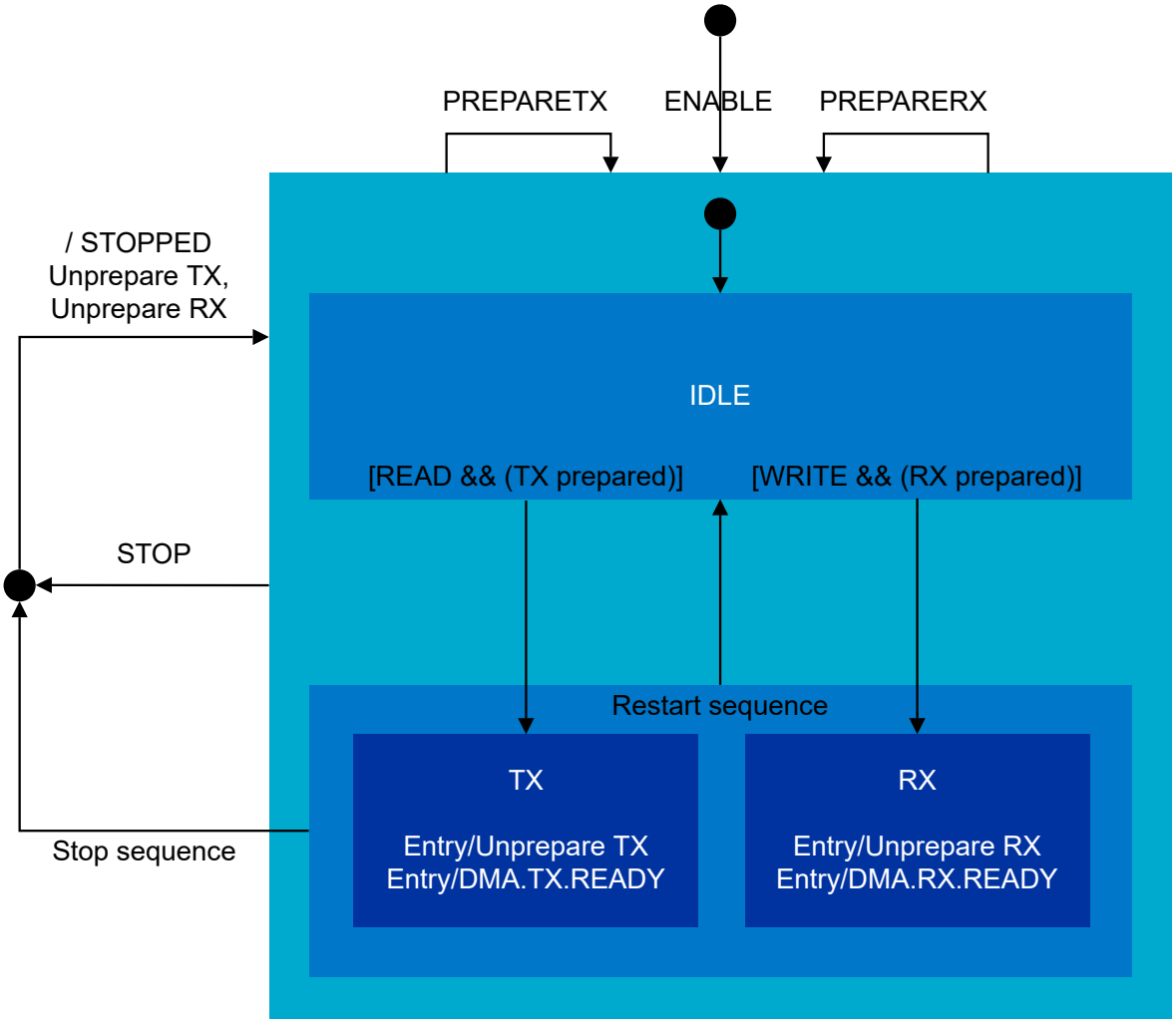


Figure 102: TWIS state machine

The following table contains descriptions of the symbols used in the state machine.

Symbol	Type	Description
ENABLE	Register	TWIS enabled via the ENABLE register.
PREPARETX	Task	The TASKS_PREPARETX task was triggered.
STOP	Task	The TASKS_STOP task was triggered.
PREPARERX	Task	The TASKS_PREPARERX task was triggered.
STOPPED	Event	The EVENTS_STOPPED event was generated.
DMA.RX.READY	Event	The EVENTS_DMA.RX.READY event was generated.
DMA.TX.READY	Event	The EVENTS_DMA.TX.READY event was generated.
TX prepared	Internal	Internal flag indicating that a TASKS_PREPARETX task was triggered.
RX prepared	Internal	Internal flag indicating that a TASKS_PREPARERX task was triggered.
Unprepare TX	Internal	Clears the TX prepared flag until the next TASKS_PREPARETX task.
Unprepare RX	Internal	Clears the RX prepared flag until the 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 48: TWI slave state machine symbols

TWIS supports clock stretching. In order to use this feature, the controller must also support clock stretching for the feature to execute properly. TWIS operates in a low-power mode while waiting for the TWI controller to initiate a transfer. As long as TWIS is not addressed, it will remain in this mode.

For TWIS to run correctly, PSEL.SCL, PSEL.SDA, CONFIG, and the ADDRESS[n] registers must be configured, the SCL and SDA lines must both be high, before enabling TWIS through the ENABLE register. Similarly, changing these settings must be performed while TWIS is disabled. Failing to do so may result in unpredictable behavior.

8.19.2 Shared resources

The TWIS peripheral shares registers and other resources with peripherals that have the same ID as TWIS. Before TWIS can be configured and used, all peripherals that have the same ID as TWIS must be disabled.

Disabling a peripheral with the same ID as TWIS will not reset any shared TWIS registers. Configure all TWIS registers to ensure they operate correctly.

See the Instantiation table in Instantiation on page 141 for details on peripherals and their IDs.

8.19.3 EasyDMA

TWIS implements EasyDMA for accessing RAM without CPU involvement.

TWIS implements the EasyDMA channels found in the following table.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 49: TWIS EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 19.

The STOPPED event indicates that EasyDMA is finished accessing the buffer in RAM.

8.19.4 Read command response

Before TWIS can respond to a read command, it must be configured and enabled in the ENABLE register. When enabled, TWIS is in the IDLE state.

A read command is started when TWIM generates a start condition on the TWI bus. This is followed by clocking out the address and setting the READ/WRITE bit to `1` (READ=1, WRITE=0). The READ/WRITE bit is followed by an ACK/NACK bit (ACK = 0, NACK = 1) response from the TWIS.

TWIS can listen for two addresses at a time. This is configured in the ADDRESS registers and the CONFIG register.

TWIS only acknowledges (ACK) the read command if the address presented by the controller matches one of the addresses the target is configured to listen for. TWIS will generate a READ event when it acknowledges the read command.

TWIS only detects 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 controller's clock until the PREPARETX task is triggered and the internal **`TX prepared`** flag is set.

TWIS will generate the EVENTS_DMA.TX.READY event and clear the **`TX prepared`** flag when it enters the TX state. In this state, TWIS will send the data bytes found in the transmit buffer to the controller using the controller's clock.

TWIS returns to the IDLE state if the TWIS receives a restart command when it is in the TX state.

TWIS is stopped when it receives the stop condition from TWIM. A STOPPED event will be generated when the transaction has stopped. TWIS will clear the **`TX prepared`** flag 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 TWIM forces TWIS to send more than TXD.MAXCNT bytes, the target will send the byte specified in the ORC register to the controller 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 EVENTS_DMA.TX.READY 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 return to the IDLE state when it has stopped, see Terminate an ongoing TWI transaction on page 458.

Each byte sent from TWIS will be followed by an ACK/NACK bit sent from the controller. TWIM will generate a NACK following the last byte that it wants to receive to tell the target to release the bus allowing TWIM to 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.

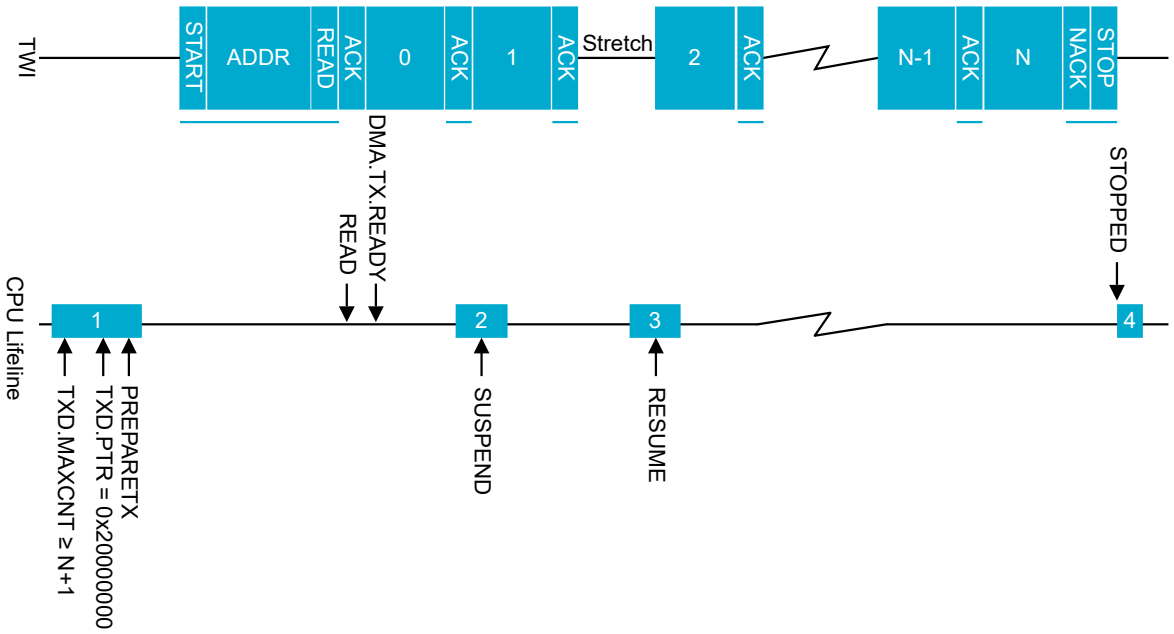


Figure 103: TWIS responding to a read command

8.19.5 Write command response

Before TWIS can respond to a write command, TWIS must be configured and enabled in the ENABLE register. When enabled, TWIS is in the IDLE state.

A write command is started when TWIM generates a start condition on the TWI bus. This is followed by clocking out the address and setting the READ/WRITE bit to 0 (READ = 1, WRITE = 0). The READ/WRITE bit is followed by an ACK/NACK bit (ACK = 0, NACK = 1) response from the TWIS.

TWIS can listen for two addresses at a time. This is configured in the ADDRESS registers and the CONFIG register.

TWIS only acknowledges (ACK) the write command if the address presented by the controller matches one of the addresses the target is configured to listen for. TWIS will generate a WRITE event when it acknowledges the write command.

TWIS only detects 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 start stretching the master's clock after the first data byte, not allowing the master to send the stop condition. Clock is stretched until the PREPARERX task is triggered and the internal **RX prepared** flag is set.

TWIS will generate the EVENTS_DMA.RX.READY event and clear the internal **RX prepared** flag when it enters the RX state. In this state, TWIS will be able to receive the bytes sent by the TWIM.

TWIS returns 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 TWIM. A STOPPED event will be generated when the transaction has stopped. TWIS will clear the internal **RX prepared** flag 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 can only receive as many bytes as specified in the RXD.MAXCNT register. If TWIM tries to send more bytes to TWIS than it

can receive, the extra bytes are discarded and NACKed by the target. 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 EVENTS_DMA.RX.READY 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 return to the IDLE state when it has stopped, see Terminate an ongoing TWI transaction on page 458.

TWIS will generate an ACK after every byte received from the controller. 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.

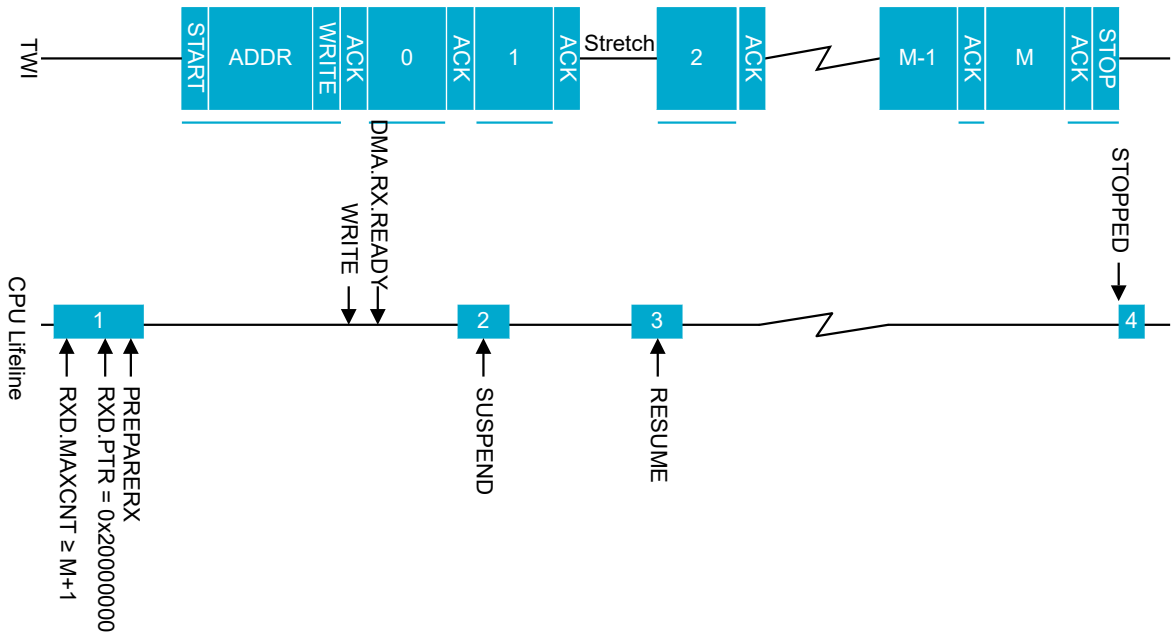


Figure 104: TWIS responding to a write command

8.19.6 TWI controller repeated start sequence

A repeated start sequence is where the TWI controller writes two bytes to TWIS, followed by reading four bytes from the target. This is shown in the following figure.

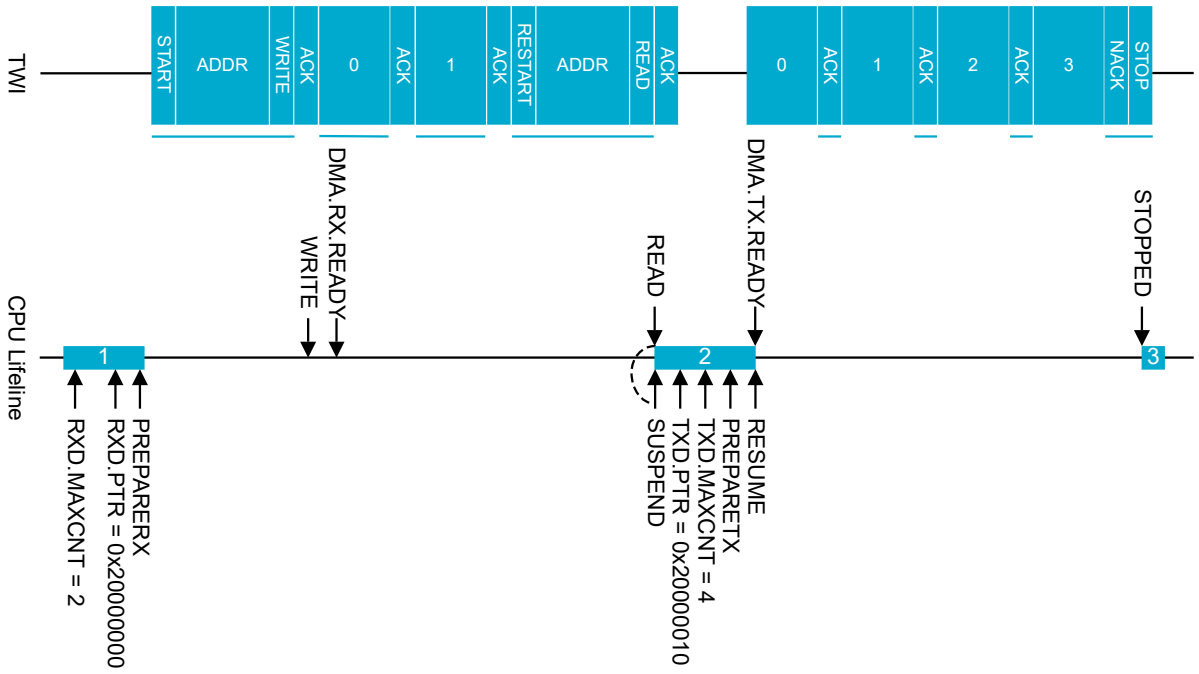


Figure 105: Repeated start sequence

In this example, the receiver does not know in advance what the controller wants to read. This information is in the first two received bytes of the write in the repeated start sequence. For the CPU to process the received data before TWIS replies to the read command, the SUSPEND task is triggered. This is enabled through 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.

8.19.7 Terminate an ongoing TWI transaction

In some situations, an ongoing transaction must be terminated. This can happen when the external TWI controller is not responding correctly, for example.

To stop an ongoing transaction, trigger the STOP task. A STOPPED event will be generated when TWIS stops. It is not dependent on the STOP condition being generated on the TWI bus. TWIS will release the bus when it has stopped and returns to its IDLE state.

8.19.8 Low power

When the peripheral is not needed, stop and disable TWIS for lowest possible power consumption.

When the STOP task is sent, the software must wait until the STOPPED event is received before disabling the peripheral through the ENABLE register. If the peripheral is already stopped, the STOP task is not needed.

8.19.9 Target 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 when TWIS is enabled, and retained while the device is in System ON mode. When the peripheral is disabled, the pins function as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. Only configure PSEL.SCL and PSEL.SDA when TWIS is disabled.

When in System OFF mode or when TWIS is disabled, the TWIS pins must be configured in the GPIO peripheral as described in the following table to secure correct signal levels.

Only one peripheral can be assigned to drive a GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWIS signal	TWIS 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 50: GPIO configuration before enabling peripheral

8.19.10 Registers

Instances

Instance	Domain	Base address	Description
TWIS20	GLOBAL	0x400C6000	Two-wire interface target TWIS20
TWIS21	GLOBAL	0x400C7000	Two-wire interface target TWIS21
TWIS22	GLOBAL	0x400C8000	Two-wire interface target TWIS22

Configuration

Instance	Domain	Configuration
TWIS20	GLOBAL	Use GPIO port P1
TWIS21	GLOBAL	Use GPIO port P1
TWIS22	GLOBAL	Use GPIO port P1

Register overview

Register	Offset	Description
TASKS_STOP	0x004	Stop TWI transaction
TASKS_SUSPEND	0x00C	Suspend TWI transaction
TASKS_RESUME	0x010	Resume TWI transaction
TASKS_PREPARERX	0x020	Prepare the TWI slave to respond to a write command
TASKS_PREPARETX	0x024	Prepare the TWI slave to respond to a read command
TASKS_DMA.RX.ENABLEMATCH[n]	0x030	Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.RX.DISABLEMATCH[n]	0x040	Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
SUBSCRIBE_STOP	0x084	Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x08C	Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x090	Subscribe configuration for task RESUME
SUBSCRIBE_PREPARERX	0x0A0	Subscribe configuration for task PREPARERX
SUBSCRIBE_PREPARETX	0x0A4	Subscribe configuration for task PREPARETX
SUBSCRIBE_DMA.RX.ENABLEMATCH[n]	0x0B0	Subscribe configuration for task ENABLEMATCH[n]
SUBSCRIBE_DMA.RX.DISABLEMATCH[n]	0x0C0	Subscribe configuration for task DISABLEMATCH[n]
EVENTS_STOPPED	0x104	TWI stopped
EVENTS_ERROR	0x114	TWI error
EVENTS_WRITE	0x13C	Write command received
EVENTS_READ	0x140	Read command received
EVENTS_DMA.RX.END	0x14C	Generated after all MAXCNT bytes have been transferred

Register	Offset	Description
EVENTS_DMA.RX.READY	0x150	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.RX.BUSERROR	0x154	An error occured during the bus transfer.
EVENTS_DMA.RX.MATCH[n]	0x158	Pattern match is detected on the DMA data bus.
EVENTS_DMA.TX.END	0x168	Generated after all MAXCNT bytes have been transferred
EVENTS_DMA.TX.READY	0x16C	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.TX.BUSERROR	0x170	An error occured during the bus transfer.
PUBLISH_STOPPED	0x184	Publish configuration for event STOPPED
PUBLISH_ERROR	0x194	Publish configuration for event ERROR
PUBLISH_WRITE	0x1BC	Publish configuration for event WRITE
PUBLISH_READ	0x1C0	Publish configuration for event READ
PUBLISH_DMA.RX.END	0x1CC	Publish configuration for event END
PUBLISH_DMA.RX.READY	0x1D0	Publish configuration for event READY
PUBLISH_DMA.RX.BUSERROR	0x1D4	Publish configuration for event BUSERROR
PUBLISH_DMA.RX.MATCH[n]	0x1D8	Publish configuration for event MATCH[n]
PUBLISH_DMA.TX.END	0x1E8	Publish configuration for event END
PUBLISH_DMA.TX.READY	0x1EC	Publish configuration for event READY
PUBLISH_DMA.TX.BUSERROR	0x1F0	Publish configuration for event BUSERROR
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
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.
PSEL.SCL	0x600	Pin select for SCL signal
PSEL.SDA	0x604	Pin select for SDA signal
DMA.RX.PTR	0x704	RAM buffer start address
DMA.RX.MAXCNT	0x708	Maximum number of bytes in channel buffer
DMA.RX.AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.RX.TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.RX.BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.
DMA.RX.MATCH.CONFIG	0x724	Configure individual match events
DMA.RX.MATCH.CANDIDATE[n]	0x728	The data to look for - any match will trigger the MATCH[n] event, if enabled.
DMA.TX.PTR	0x73C	RAM buffer start address
DMA.TX.MAXCNT	0x740	Maximum number of bytes in channel buffer
DMA.TX.AMOUNT	0x744	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.TX.TERMINATEONBUSERROR	0x754	Terminate the transaction if a BUSERROR event is detected.
DMA.TX.BUSERRORADDRESS	0x758	Address of transaction that generated the last BUSERROR event.

8.19.10.1 TASKS_STOP

Address offset: 0x004

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

8.19.10.2 TASKS_SUSPEND

Address offset: 0x00C

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

8.19.10.3 TASKS_RESUME

Address offset: 0x010

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

8.19.10.4 TASKS_PREPARERX

Address offset: 0x020

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

8.19.10.5 TASKS_PREPARETX

Address offset: 0x024

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

8.19.10.6 TASKS_DMA

Peripheral tasks.

8.19.10.6.1 TASKS_DMA.RX

Peripheral tasks.

8.19.10.6.1.1 TASKS_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x030 + (n × 0x4)

Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG 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	ENABLEMATCH			Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.19.10.6.1.2 TASKS_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG 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	DISABLEMATCH			Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.19.10.7 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 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.8 SUBSCRIBE_SUSPEND

Address offset: 0x08C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.9 SUBSCRIBE_RESUME

Address offset: 0x090

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.10 SUBSCRIBE_PREPARERX

Address offset: 0x0A0

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task PREPARERX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.11 SUBSCRIBE_PREPARETX

Address offset: 0x0A4

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that task PREPARETX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.12 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.19.10.12.1 SUBSCRIBE_DMA.RX

Subscribe configuration for tasks

8.19.10.12.1.1 SUBSCRIBE_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task ENABLEMATCH[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		[0..255]	DPPI channel that task ENABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.12.1.2 SUBSCRIBE_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task DISABLEMATCH[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		[0..255]	DPPI channel that task DISABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.19.10.13 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

8.19.10.14 EVENTS_ERROR

Address offset: 0x114

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

8.19.10.15 EVENTS_WRITE

Address offset: 0x13C

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

8.19.10.16 EVENTS_READ

Address offset: 0x140

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

8.19.10.17 EVENTS_DMA

Peripheral events.

8.19.10.17.1 EVENTS_DMA.RX

Peripheral events.

8.19.10.17.1.1 EVENTS_DMA.RX.END

Address offset: 0x14C

Generated after all MAXCNT bytes have been transferred

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated after all MAXCNT bytes have been transferred
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.19.10.17.1.2 EVENTS_DMA.RX.READY

Address offset: 0x150

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.19.10.17.1.3 EVENTS_DMA.RX.BUSERROR

Address offset: 0x154

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.19.10.17.1.4 EVENTS_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x158 + (n × 0x4)

Pattern match is detected on the DMA data bus.

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

8.19.10.17.2 EVENTS_DMA.TX

Peripheral events.

8.19.10.17.2.1 EVENTS_DMA.TX.END

Address offset: 0x168

Generated after all MAXCNT bytes have been transferred

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

8.19.10.17.2.2 EVENTS_DMA.TX.READY

Address offset: 0x16C

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	READY																																	
			NotGenerated	0																															
			Generated	1																															

8.19.10.17.2.3 EVENTS_DMA.TX.BUSERROR

Address offset: 0x170

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.19.10.18 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 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.19 PUBLISH_ERROR

Address offset: 0x194

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.20 PUBLISH_WRITE

Address offset: 0x1BC

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event WRITE will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.21 PUBLISH_READ

Address offset: 0x1C0

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event READ will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22 PUBLISH_DMA

Publish configuration for events

8.19.10.22.1 PUBLISH_DMA.RX

Publish configuration for events

8.19.10.22.1.1 PUBLISH_DMA.RX.END

Address offset: 0x1CC

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.1.2 PUBLISH_DMA.RX.READY

Address offset: 0x1D0

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.1.3 PUBLISH_DMA.RX.BUSERROR

Address offset: 0x1D4

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.1.4 PUBLISH_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x1D8 + (n × 0x4)

Publish configuration for event MATCH[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		[0..255]	DPPI channel that event MATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.2 PUBLISH_DMA.TX

Publish configuration for events

8.19.10.22.2.1 PUBLISH_DMA.TX.END

Address offset: 0x1E8

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.2.2 PUBLISH_DMA.TX.READY

Address offset: 0x1EC

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.22.2.3 PUBLISH_DMA.TX.BUSERROR

Address offset: 0x1F0

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.19.10.23 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				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	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
C-F	RW	DMA_RX_MATCH[i]_DMA_RX_ENABLEMATCH[(i+1)%4] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.ENABLEMATCH[(i+1)%4] Allows daisy-chaining match events.
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G-J	RW	DMA_RX_MATCH[i]_DMA_RX_DISABLEMA (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.DISABLEMATCH[n]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

8.19.10.24 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							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																															
A	RW	STOPPED																																	
			Disabled	0																															
			Enabled	1																															
B	RW	ERROR																																	
			Disabled	0																															
			Enabled	1																															
C	RW	WRITE																																	
			Disabled	0																															
			Enabled	1																															
D	RW	READ																																	
			Disabled	0																															
			Enabled	1																															
E	RW	DMARXEND																																	
			Disabled	0																															
			Enabled	1																															
F	RW	DMARXREADY																																	
			Disabled	0																															
			Enabled	1																															
G	RW	DMARXBUSERROR																																	
				When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.																																
			Disabled	0																															
			Enabled	1																															
H-K	RW	DMARXMATCH[i] (i=0..3)																																	
			Disabled	0																															
			Enabled	1																															
L	RW	DMATXEND																																	
			Disabled	0																															
			Enabled	1																															
M	RW	DMATXREADY																																	
			Disabled	0																															
			Enabled	1																															
N	RW	DMATXBUSERROR																																	
				When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.																																
			Disabled	0																															
			Enabled	1																															

8.19.10.25 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				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 W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	WRITE			Write '1' to enable interrupt for event WRITE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	READ			Write '1' to enable interrupt for event READ
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	DMARXEND			Write '1' to enable interrupt for event DMARXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	DMARXREADY			Write '1' to enable interrupt for event DMARXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1S	DMARXBUSERROR			Write '1' to enable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H-K	RW W1S	DMARXMATCH[i] (i=0..3)			Write '1' to enable interrupt for event DMARXMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1S	DMATXEND			Write '1' to enable interrupt for event DMATXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1S	DMATXREADY			Write '1' to enable interrupt for event DMATXREADY
			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				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
N	RW W1S	DMATXBUSERROR			Write '1' to enable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.19.10.26 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				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 W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	WRITE			Write '1' to disable interrupt for event WRITE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	READ			Write '1' to disable interrupt for event READ
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	DMARXEND			Write '1' to disable interrupt for event DMARXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	DMARXREADY			Write '1' to disable interrupt for event DMARXREADY
			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				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
G	RW W1C	DMARXBUSERROR			Write '1' to disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H-K	RW W1C	DMARXMATCH[i] (i=0..3)			Write '1' to disable interrupt for event DMARXMATCH[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW W1C	DMATXEND			Write '1' to disable interrupt for event DMATXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW W1C	DMATXREADY			Write '1' to disable interrupt for event DMATXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW W1C	DMATXBUSERROR			Write '1' to disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.19.10.27 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 W1C	OVERFLOW			RX buffer overflow detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred
B	RW W1C	DNACK			NACK sent after receiving a data byte
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW W1C	OVERREAD			TX buffer over-read detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred

8.19.10.28 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

8.19.10.29 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

8.19.10.30 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

8.19.10.31 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

8.19.10.32 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.

8.19.10.33 PSEL.SCL

Address offset: 0x600

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 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	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.19.10.34 PSEL.SDA

Address offset: 0x604

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 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	RW	PIN		[0..31]	Pin number
B	RW	PORT		[0..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.19.10.35 DMA.RX.PTR

Address offset: 0x704

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.19.10.36 DMA.RX.MAXCNT

Address offset: 0x708

Maximum number of bytes in channel buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0xFFFF]	Maximum number of bytes in channel buffer

8.19.10.37 DMA.RX.AMOUNT

Address offset: 0x70C

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.19.10.38 DMA.RX.TERMINATEONBUSERROR

Address offset: 0x71C

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.19.10.39 DMA.RX.BUSERRORADDRESS

Address offset: 0x720

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	ADDRESS			

8.19.10.40 DMA.RX.MATCH

Registers to control the behavior of the pattern matcher engine

8.19.10.40.1 DMA.RX.MATCH.CONFIG

Address offset: 0x724

Configure individual match events

Bit number	31	30	29	28	27	26	25	24	23	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-D	RW	ENABLE[i] (i=0..3)			Enable match filter i
			Disabled	0	Match filter disabled
			Enabled	1	Match filter enabled
E-H	RW	ONESHOT[i] (i=0..3)			Configure match filter i as one-shot or sticky One-shot match filters can be used together with shortcuts to check for continuous data sequences by disabling the filter if the next data is not a match. **Note:** The presence of these shorts depends on the configuration of the peripheral integrating this EasyDMA.
			Continuous	0	Match filter stays enabled until disabled by task
			Oneshot	1	Match filter stays enabled until next data word is received

8.19.10.40.2 DMA.RX.MATCH.CANDIDATE[n] (n=0..3)

Address offset: 0x728 + (n × 0x4)

The data to look for - any match will trigger the MATCH[n] event, if enabled.

Note: This register can be updated while a transfer is in progress, but the new value will not take effect until a match has been found or the transfer is done. That makes it possible to write a new set of match words which will be searched for immediately after the event triggers.

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

8.19.10.41 DMA.TX.PTR

Address offset: 0x73C

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.19.10.42 DMA.TX.MAXCNT

Address offset: 0x740

Maximum number of bytes in channel 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 channel buffer

8.19.10.43 DMA.TX.AMOUNT

Address offset: 0x744

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.19.10.44 DMA.TX.TERMINATEONBUSERROR

Address offset: 0x754

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.19.10.45 DMA.TX.BUSERRORADDRESS

Address offset: 0x758

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	ADDRESS			

8.20 UARTE — Universal asynchronous receiver/transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA peripheral (UARTE) provides a full-duplex, asynchronous serial communication interface with hardware flow control.

The main features of UARTE are the following:

- Full-duplex operation
- EasyDMA direct transfer to and from RAM
- Individual selection of I/O pins
- Slow instances with up to 1 Mbps baud rate
- Optional even and odd parity bit checking and generation
- One or two stop bits
- Configurable data frame size: 4 bit to 9 bit
- 9-bit mode support with address matching in RX
- Automatic hardware flow control
- Supports return to the IDLE state between transactions (when using HW flow control)
- Interrupt generation after programmable timeout
- Compare match filter for generating events or interrupts

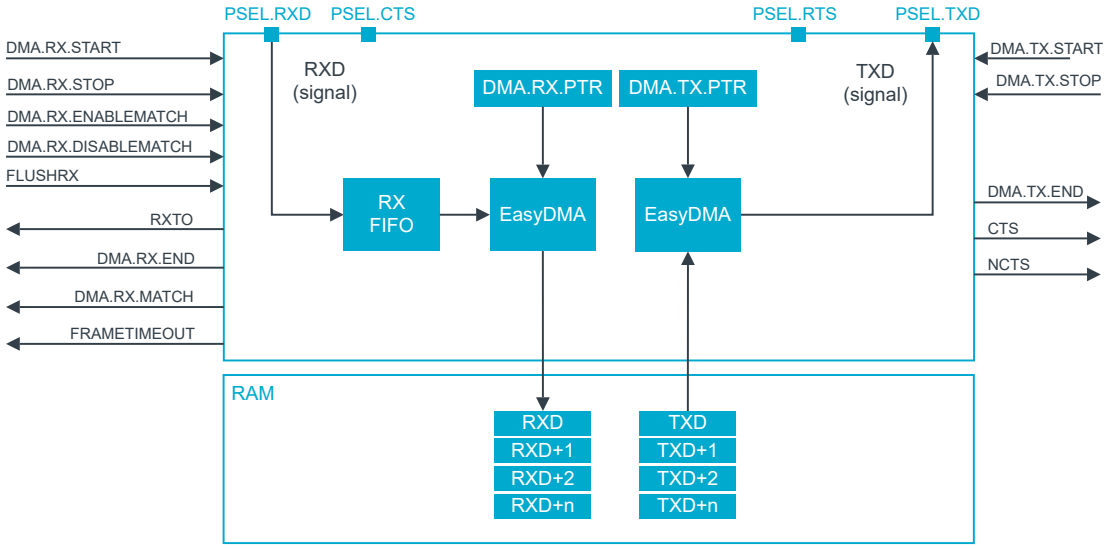


Figure 106: UARTE configuration

Note: The external crystal oscillator must be enabled to obtain sufficient clock accuracy for stable communication. See CLOCK — Clock control on page 54 for more information.

8.20.1 Baudrate

The UART baudrate defines the speed at which data is transmitted over the UART interface, measured in bits per second (bps).

The BAUDRATE register lists a set of precalculated values for the most common baudrates and 16 MHz PCLK. For the high speed instances (PCLK > 16 MHz), the baudrate is calculated as follows:

$$\mathrm{BAUDRATE} = 2^{12} \times \left\lfloor \frac{2^{20}}{\mathrm{round}\left(\frac{\mathrm{f\_PCLK}}{\mathrm{desired\_baudrate}}\right)} \right\rfloor$$

Figure 107: UARTE baudrate

- BAUDRATE is the value to be used in the UARTE BAUDRATE register.
- f_PCLK is the peripheral clock frequency for the UARTE instance, as defined in Instances on page 489.
- desired_baudrate is the desired baudrate in bits per second, such as 9600 or 115200.

8.20.2 EasyDMA

UARTE implements EasyDMA for reading and writing to and from RAM.

If the DMA.TX.PTR and the DMA.RX.PTR are not pointing to the RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 11 for more information about each memory region.

The DMA.RX.PTR, DMA.TX.PTR, DMA.RX.MAXCNT, and DMA.TX.MAXCNT registers are double-buffered. They can be updated and prepared for the next reception or transmission immediately after having received the DMA.RX.READY or DMA.TX.READY events.

The DMA.RX.END and DMA.TX.END events indicate that the EasyDMA is finished accessing the RX or TX buffer in RAM.

For detailed information regarding the use of EasyDMA, see EasyDMA on page 19.

8.20.3 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to DMA.TX.PTR, and the number of bytes to transmit from the RAM buffer to DMA.TX.MAXCNT. The UARTE transmission is started by triggering the DMA.TX.START task.

After each byte has been sent over the TXD line, a TXDRDY event is generated.

When then bytes have been transmitted, the DMA.TX.END event is generated.

A UARTE transmission sequence is stopped by triggering the DMA.TX.STOP task. A TXSTOPPED event will be generated when the UARTE transmitter has stopped.

If the DMA.TX.END event has not been generated when the UARTE transmitter stops, UARTE will generate the DMA.TX.END event explicitly even though all bytes specified in the DMA.TX.MAXCNT register have not been transmitted.

If flow control is enabled in 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 finish transmitting before the transmission is suspended.

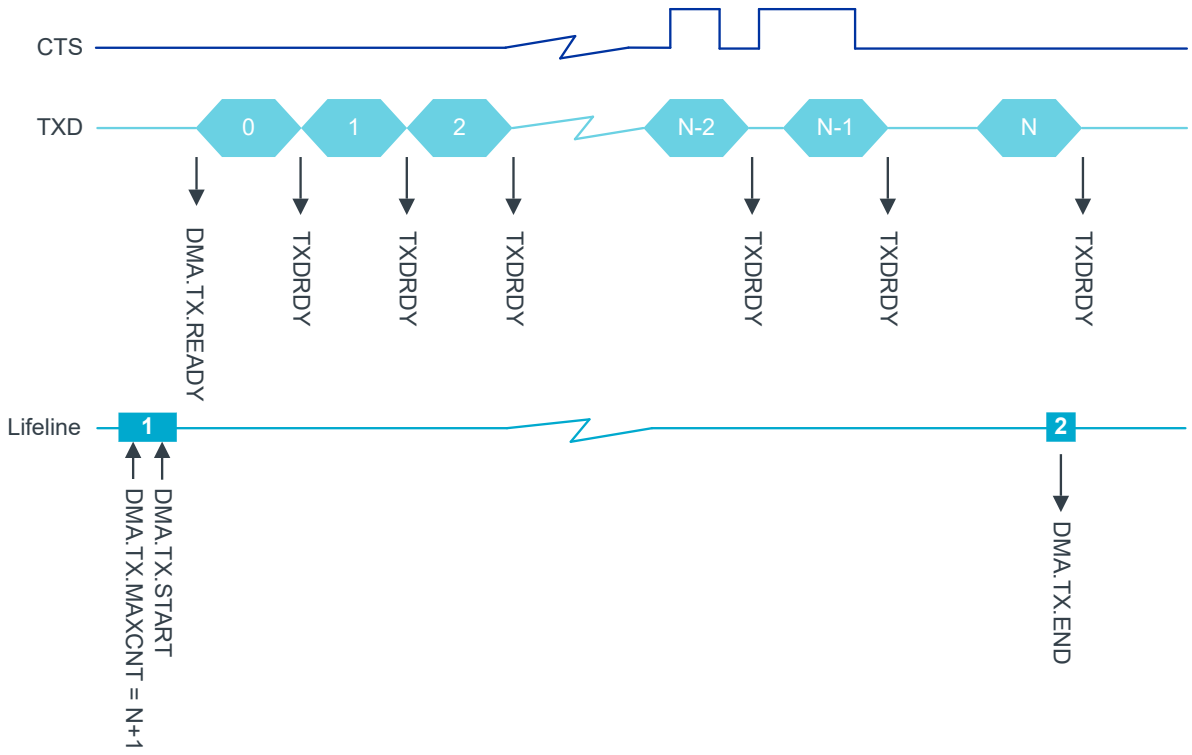


Figure 108: UARTE transmission

The UARTE transmitter is least active when it is stopped, consuming the least amount of energy. This is before it is started via DMA.TX.START, or after it has been stopped via DMA.TX.STOP and the TXSTOPPED event has been generated. See POWER — Power control on page 79 for more information about power modes.

8.20.4 Reception

The UARTE receiver is started by triggering the DMA.RX.START 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 DMA.RX.PTR register. The DMA.RX.PTR register is double-buffered and can be updated and prepared for the next DMA.RX.START task immediately after the EVENTS_DMA.RX.READY event is generated. The size of the RX buffer is specified in the DMA.RX.MAXCNT register. UARTE generates an DMA.RX.END 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 RAM.

The DMA.RX.AMOUNT register can be queried following an DMA.RX.END event to see how many new bytes have been transferred to the RX buffer in RAM since the previous DMA.RX.END event.

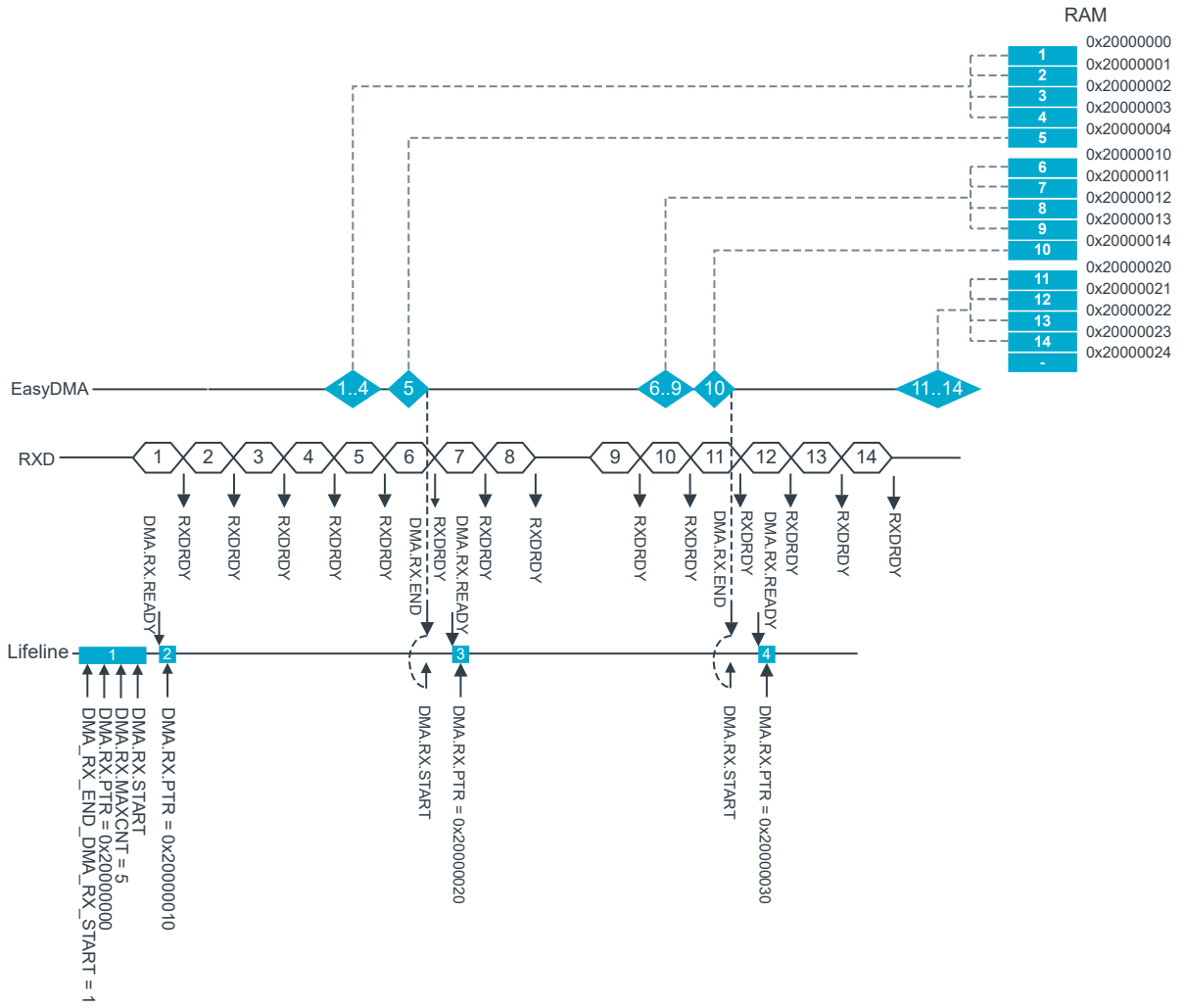


Figure 109: UARTE reception

The UARTE receiver is stopped by triggering the DMA.RX.STOP task. The RXTO and DMA.RX.END events are generated when UARTE has stopped.

Stopping the UARTE with a RX buffer that is empty or not completely filled up will also cause the DMA.RX.END event to be generated.

The number of bytes stored in the RX buffer can be found by reading the DMA.RX.AMOUNT register following the DMA.RX.END event.

To stop UARTE, trigger the DMA.RX.STOP task. After the STOP task is triggered, UARTE can receive up to four bytes during the timeout period. The data received during the timeout period will be written to DMA.RX.PTR, and limited by DMA.RX.MAXCNT registers. MAXCNT should be set to at least four bytes to allow for this, while following the DMA.RX.END event, the DMA.RX.AMOUNT register will indicate the actual number of bytes received.

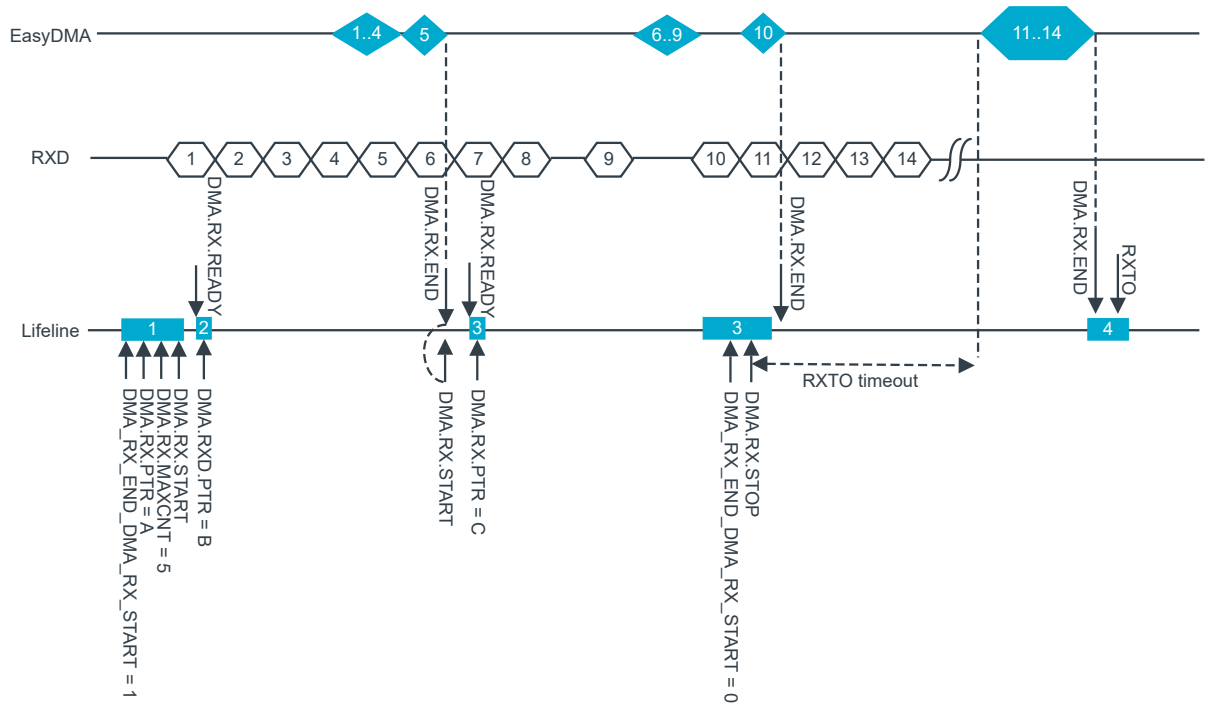


Figure 110: UARTE reception with forced stop through DMA.RX.STOP

If hardware flow control is enabled in the HWFC field in the CONFIG register, the RTS signal will be deactivated when the receiver is stopped via the DMA.RX.STOP task, or when UARTE can only receive three 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 UARTE is only able to receive three additional bytes in its internal RX FIFO. Data received when the internal RX FIFO is full, will be lost.

The UARTE receiver is least active when it is stopped, consuming the least amount of energy. This is before it is started via DMA.RX.START, or after it has been stopped via DMA.RX.STOP and the RXTO event has been generated. See POWER — Power control on page 79 for more information about power modes.

8.20.5 Data frame size

UARTE implements a configurable data frame size of 4 bits to 9 bits and is set in the register CONFIG on page 511. If a value greater than 9 or less than 4 is written to this register, the frame size will be set to 8 bits and the register will read back a value of 8.

When UARTE is used with the 9 bit frame size, a 9th bit is added after the MSB of the 8 bit data frame, and before the parity and stop bits, as shown in the following figure. This bit indicates if the 8 bit data is an address or data. If the 9th bit is `1`, the 8 bit data is interpreted as an address. If the 9th bit is `0`, the 8 bit data is interpreted as data.



Figure 111: UARTE frame

When UART is in RX configured with a 9 bit frame size, all frames are ignored until a frame with the address bit set is received, and the 8 remaining bits of the data frame matches the address set in the register ADDRESS on page 512. The frames following the matching address are received as an 8 bit data frame until the next frame where the address bit set is received, the address bit is not stored. If the address does not match ADDRESS, the following frames are ignored.

If the parity bit is enabled, the address bit is not included in the parity calculation.

When UART TX is started, the first byte in the buffer read by EasyDMA is treated as an address, and transmitted with the address bit set to `1`. The next bits in the buffer are treated as data and transmitted with the address bit set to `0`.

When UARTE uses a data frame size less than an 8 bits, the data is trimmed from an 8 bit frame size in the RAM buffer for TX, and padded before being stored in the RAM buffer for RX. The ENDIAN field in the register CONFIG defines if the data is trimmed from MSB or LSB of the 8 bit buffer frame.

8.20.6 Frame timeout interrupt

UARTE can generate an event after a programmable timeout.

If enabled with the FRAMETIMEOUT field in register CONFIG on page 511, a counter starts at the start flag, and counts the number of bit periods given in register FRAMETIMEOUT on page 512. The period time of the counter is equal to the period of one bit on the UART TX line given by the BAUDRATE register. When the timeout expires, the event EVENTS_FRAMETIMEOUT is generated. The STOP task also stops the timeout counter. UARTE reception can be stopped on timeout by setting the short from the FRAMETIMEOUT event to the DMA.RX.STOP task in register SHORTS on page 504.

This feature can be used to support variable length UART transmission with no end of transmission tag. After the last UART frames are received within the configured timeout, an interrupt is generated and the data can be processed.

The following figure shows an example where the UART receives a transmission that times out after byte 12.

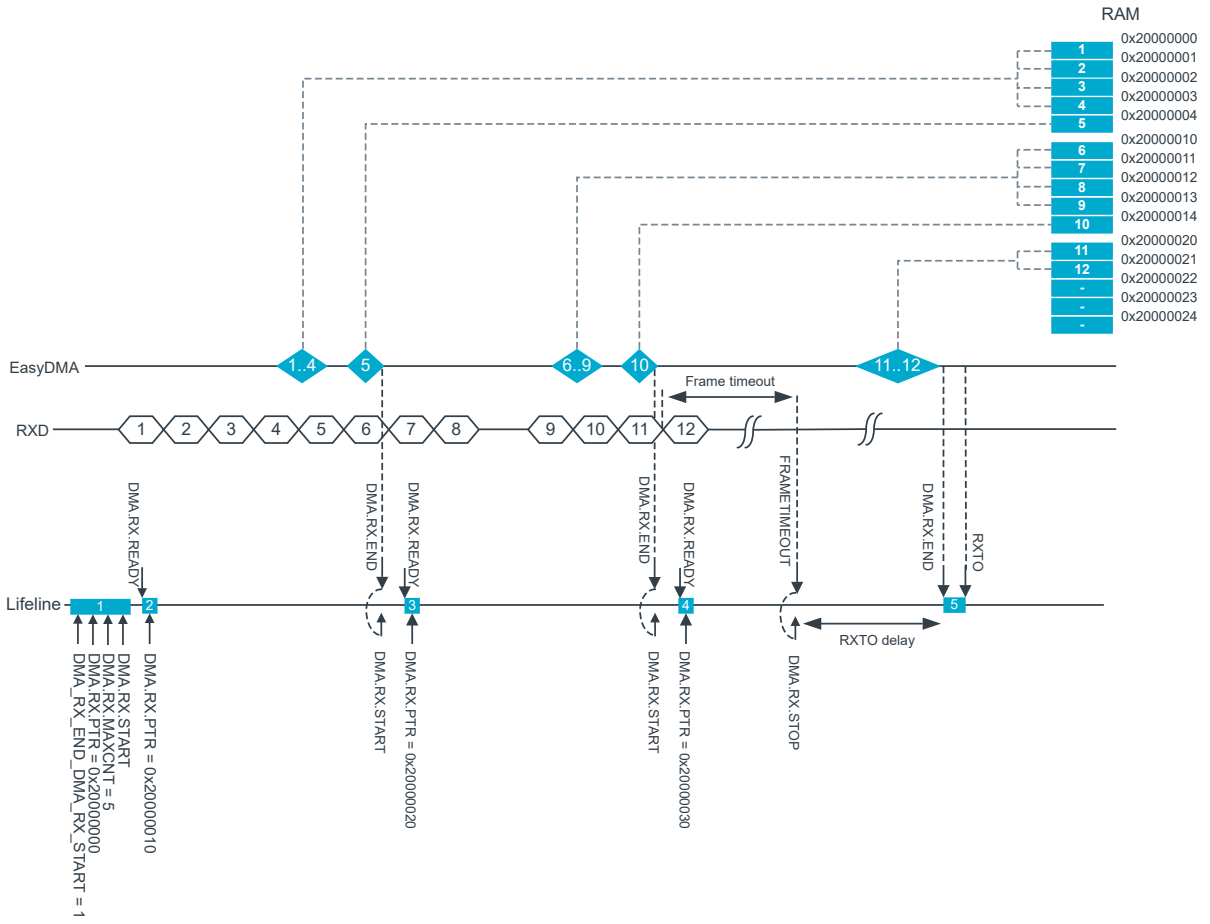


Figure 112: UARTE reception frame timeout

The minimum value of the FRAMETIMEOUT register must be set to a value larger than the configured UART frame length.

Note: Frames received after the RXTO event are discarded and not stored in memory. Reception will commence when DMA.RX.START is triggered.

8.20.7 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. 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 RAM along with any following bytes. If a framing error occurs (wrong stop bit), that byte will not be stored in RAM but the next incoming bytes will.

8.20.8 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are held active.

8.20.9 Parity and stop bit configuration

Automatic even parity generation for both transmission and reception can be configured using the register CONFIG on page 511. If odd parity is required, it can be configured using the register CONFIG on page 511. See the register description for details.

The amount of stop bits can be configured in the register CONFIG on page 511.

8.20.10 Compare match filter

UARTE has a compare match filter that can watch for a specific sequence of data. This feature is implemented in EasyDMA on the RX channel.

UARTE can generate events or interrupts when the specific data sequence is received. The event EVENTS_DMA.RX.MATCH[n] (n=0..3) on page 498 is generated when there is a match in the data stream being received. The number of MATCH can be different for each instance. See Registers on page 489 for how many MATCH events are implemented per instance.

Register DMA.RX.MATCH.CANDIDATE[n] (n=0..3) on page 515 configures the pattern for comparison. The filter is enabled with the corresponding ENABLE bit in register DMA.RX.MATCH.CONFIG on page 515. The DMA.RX.ENABLEMATCH task can be used to set that bit, and the DMA.RX.DISABLEMATCH task will clear that bit. If the ONESHOT[i] field in the CONFIG register is set, the corresponding ENABLE[i] bit will be cleared on a successful match.

For detailed information regarding the use of pattern matching in the EasyDMA engine, see EasyDMA on page 19.

8.20.11 Low power

To ensure lowest possible power consumption when the peripheral is not needed, stop and disable UARTE.

The DMA.TX.STOP and DMA.RX.STOP tasks are not always needed (the peripheral might already be stopped). If DMA.TX.STOP or DMA.RX.STOP is sent, software waits until the TXSTOPPED or RXTO event is received before disabling the peripheral through the ENABLE register.

8.20.12 Pin configuration

The RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD signals associated with UARTE are mapped to physical pins according to the configuration specified in the PSEL.n registers.

These registers and their configurations are only used when UARTE is enabled, and retained while the device is in System ON mode. The PSEL.n registers can be configured only when UARTE is disabled.

To ensure correct behavior 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 51: GPIO configuration before enabling peripheral

8.20.13 Registers

Instances

Instance	Domain	Base address	Description
UARTE20	GLOBAL	0x400C6000	Universal asynchronous receiver/transmitter UARTE20
UARTE21	GLOBAL	0x400C7000	Universal asynchronous receiver/transmitter UARTE21
UARTE22	GLOBAL	0x400C8000	Universal asynchronous receiver/transmitter UARTE22

Configuration

Instance	Domain	Configuration
UARTE20	GLOBAL	Use GPIO port P1 Supports up to 1 Mbps Timeout interrupt is included. Supports data frame sizes 4, 5, 6, 7, 8, and 9 bits. Peripheral clock frequency is 16 MHz.
UARTE21	GLOBAL	Use GPIO port P1 Supports up to 1 Mbps Timeout interrupt is included. Supports data frame sizes 4, 5, 6, 7, 8, and 9 bits. Peripheral clock frequency is 16 MHz.
UARTE22	GLOBAL	Use GPIO port P1 Supports up to 1 Mbps Timeout interrupt is included. Supports data frame sizes 4, 5, 6, 7, 8, and 9 bits. Peripheral clock frequency is 16 MHz.

Register overview

Register	Offset	Description
TASKS_FLUSHRX	0x01C	Flush RX FIFO into RX buffer
TASKS_DMA.RX.START	0x028	Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
TASKS_DMA.RX.STOP	0x02C	Stops operation using easyDMA. This does not trigger an END event.
TASKS_DMA.RX.ENABLEMATCH[n]	0x030	Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.RX.DISABLEMATCH[n]	0x040	Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
TASKS_DMA.TX.START	0x050	Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
TASKS_DMA.TX.STOP	0x054	Stops operation using easyDMA. This does not trigger an END event.
SUBSCRIBE_FLUSHRX	0x09C	Subscribe configuration for task FLUSHRX
SUBSCRIBE_DMA.RX.START	0x0A8	Subscribe configuration for task START
SUBSCRIBE_DMA.RX.STOP	0x0AC	Subscribe configuration for task STOP

Register	Offset	Description
SUBSCRIBE_DMA.RX.ENABLEMATCH[n]	0x0B0	Subscribe configuration for task ENABLEMATCH[n]
SUBSCRIBE_DMA.RX.DISABLEMATCH[n]	0x0C0	Subscribe configuration for task DISABLEMATCH[n]
SUBSCRIBE_DMA.TX.START	0x0D0	Subscribe configuration for task START
SUBSCRIBE_DMA.TX.STOP	0x0D4	Subscribe configuration for task STOP
EVENTS_CTS	0x100	CTS is activated (set low). Clear To Send.
EVENTS_NCTS	0x104	CTS is deactivated (set high). Not Clear To Send.
EVENTS_TXDRDY	0x10C	Data sent from TXD
EVENTS_RXDRDY	0x110	Data received in RXD (but potentially not yet transferred to Data RAM)
EVENTS_ERROR	0x114	Error detected
EVENTS_RXTO	0x124	Receiver timeout
EVENTS_TXSTOPPED	0x130	Transmitter stopped
EVENTS_DMA.RX.END	0x14C	Generated after EasyDMA has completed its operation.
EVENTS_DMA.RX.READY	0x150	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.RX.BUSERROR	0x154	An error occured during the bus transfer.
EVENTS_DMA.RX.MATCH[n]	0x158	Pattern match is detected on the DMA data bus.
EVENTS_DMA.TX.END	0x168	Generated after EasyDMA has completed its operation.
EVENTS_DMA.TX.READY	0x16C	Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
EVENTS_DMA.TX.BUSERROR	0x170	An error occured during the bus transfer.
EVENTS_FRAMETIMEOUT	0x174	Timed out due to bus being idle while receiving data.
PUBLISH_CTS	0x180	Publish configuration for event CTS
PUBLISH_NCTS	0x184	Publish configuration for event NCTS
PUBLISH_TXDRDY	0x18C	Publish configuration for event TXDRDY
PUBLISH_RXDRDY	0x190	Publish configuration for event RXDRDY
PUBLISH_ERROR	0x194	Publish configuration for event ERROR
PUBLISH_RXTO	0x1A4	Publish configuration for event RXTO
PUBLISH_TXSTOPPED	0x1B0	Publish configuration for event TXSTOPPED
PUBLISH_DMA.RX.END	0x1CC	Publish configuration for event END
PUBLISH_DMA.RX.READY	0x1D0	Publish configuration for event READY
PUBLISH_DMA.RX.BUSERROR	0x1D4	Publish configuration for event BUSERROR
PUBLISH_DMA.RX.MATCH[n]	0x1D8	Publish configuration for event MATCH[n]
PUBLISH_DMA.TX.END	0x1E8	Publish configuration for event END
PUBLISH_DMA.TX.READY	0x1EC	Publish configuration for event READY
PUBLISH_DMA.TX.BUSERROR	0x1F0	Publish configuration for event BUSERROR
PUBLISH_FRAMETIMEOUT	0x1F4	Publish configuration for event FRAMETIMEOUT
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
BAUDRATE	0x524	Baud rate. Accuracy depends on the HFCLK source selected.
CONFIG	0x56C	Configuration of parity, hardware flow control, framesize, and packet timeout.
ADDRESS	0x574	Set the address of the UARTE for RX when used in 9 bit data frame mode.
FRAMETIMEOUT	0x578	Set the number of UARTE bits to count before triggering packet timeout.
PSEL.TXD	0x604	Pin select for TXD signal
PSEL.CTS	0x608	Pin select for CTS signal
PSEL.RXD	0x60C	Pin select for RXD signal
PSEL.RTS	0x610	Pin select for RTS signal
DMA.RX.PTR	0x704	RAM buffer start address
DMA.RX.MAXCNT	0x708	Maximum number of bytes in channel buffer

Register	Offset	Description
DMA.RX.AMOUNT	0x70C	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.RX.TERMINATEONBUSERROR	0x71C	Terminate the transaction if a BUSERROR event is detected.
DMA.RX.BUSERRORADDRESS	0x720	Address of transaction that generated the last BUSERROR event.
DMA.RX.MATCH.CONFIG	0x724	Configure individual match events
DMA.RX.MATCH.CANDIDATE[n]	0x728	The data to look for - any match will trigger the MATCH[n] event, if enabled.
DMA.TX.PTR	0x73C	RAM buffer start address
DMA.TX.MAXCNT	0x740	Maximum number of bytes in channel buffer
DMA.TX.AMOUNT	0x744	Number of bytes transferred in the last transaction, updated after the END event. Also updated after each MATCH event.
DMA.TX.TERMINATEONBUSERROR	0x754	Terminate the transaction if a BUSERROR event is detected.
DMA.TX.BUSERRORADDRESS	0x758	Address of transaction that generated the last BUSERROR event.

8.20.13.1 TASKS_FLUSHRX

Address offset: 0x01C

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

8.20.13.2 TASKS_DMA

Peripheral tasks.

8.20.13.2.1 TASKS_DMA.RX

Peripheral tasks.

8.20.13.2.1.1 TASKS_DMA.RX.START

Address offset: 0x028

Starts operation using easyDMA to load the values. See peripheral description for operation using 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	W	START			Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
			Trigger	1	Trigger task

8.20.13.2.1.2 TASKS_DMA.RX.STOP

Address offset: 0x02C

Stops operation using easyDMA. This does not trigger an END 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	STOP			Stops operation using easyDMA. This does not trigger an END event.
			Trigger	1	Trigger task

8.20.13.2.1.3 TASKS_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x030 + (n × 0x4)

Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG 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	ENABLEMATCH			Enables the MATCH[n] event by setting the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.20.13.2.1.4 TASKS_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x040 + (n × 0x4)

Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG 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	DISABLEMATCH			Disables the MATCH[n] event by clearing the ENABLE[n] bit in the CONFIG register.
			Trigger	1	Trigger task

8.20.13.2.2 TASKS_DMA.TX

Peripheral tasks.

8.20.13.2.2.1 TASKS_DMA.TX.START

Address offset: 0x050

Starts operation using easyDMA to load the values. See peripheral description for operation using 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	W	START			Starts operation using easyDMA to load the values. See peripheral description for operation using easyDMA.
			Trigger	1	Trigger task

8.20.13.2.2.2 TASKS_DMA.TX.STOP

Address offset: 0x054

Stops operation using easyDMA. This does not trigger an END 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	STOP			Stops operation using easyDMA. This does not trigger an END event.
			Trigger	1	Trigger task

8.20.13.3 SUBSCRIBE_FLUSHRX

Address offset: 0x09C

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		[0..255]	DPPI channel that task FLUSHRX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4 SUBSCRIBE_DMA

Subscribe configuration for tasks

8.20.13.4.1 SUBSCRIBE_DMA.RX

Subscribe configuration for tasks

8.20.13.4.1.1 SUBSCRIBE_DMA.RX.START

Address offset: 0x0A8

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4.1.2 SUBSCRIBE_DMA.RX.STOP

Address offset: 0x0AC

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4.1.3 SUBSCRIBE_DMA.RX.ENABLEMATCH[n] (n=0..3)

Address offset: 0x0B0 + (n × 0x4)

Subscribe configuration for task ENABLEMATCH[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		[0..255]	DPPI channel that task ENABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4.1.4 SUBSCRIBE_DMA.RX.DISABLEMATCH[n] (n=0..3)

Address offset: 0x0C0 + (n × 0x4)

Subscribe configuration for task DISABLEMATCH[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		[0..255]	DPPI channel that task DISABLEMATCH[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4.2 SUBSCRIBE_DMA.TX

Subscribe configuration for tasks

8.20.13.4.2.1 SUBSCRIBE_DMA.TX.START

Address offset: 0x0D0

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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.4.2.2 SUBSCRIBE_DMA.TX.STOP

Address offset: 0x0D4

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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.20.13.5 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

8.20.13.6 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 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

8.20.13.7 EVENTS_TXDRDY

Address offset: 0x10C

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 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

8.20.13.8 EVENTS_RXDRDY

Address offset: 0x110

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 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

8.20.13.9 EVENTS_ERROR

Address offset: 0x114

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

8.20.13.10 EVENTS_RXTO

Address offset: 0x124

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

8.20.13.11 EVENTS_TXSTOPPED

Address offset: 0x130

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																															
A	RW	EVENTS_TXSTOPPED																																	
			NotGenerated	0																															
			Generated	1																															

8.20.13.12 EVENTS_DMA

Peripheral events.

8.20.13.12.1 EVENTS_DMA.RX

Peripheral events.

8.20.13.12.1.1 EVENTS_DMA.RX.END

Address offset: 0x14C

Generated after EasyDMA has completed its 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																															
A	RW	END																																	
			NotGenerated	0																															
			Generated	1																															

8.20.13.12.1.2 EVENTS_DMA.RX.READY

Address offset: 0x150

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	READY																																	
			NotGenerated	0																															
			Generated	1																															

8.20.13.12.1.3 EVENTS_DMA.RX.BUSERROR

Address offset: 0x154

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.12.1.4 EVENTS_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x158 + (n × 0x4)

Pattern match is detected on the DMA data bus.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	MATCH			Pattern match is detected on the DMA data bus.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.12.2 EVENTS_DMA.TX

Peripheral events.

8.20.13.12.2.1 EVENTS_DMA.TX.END

Address offset: 0x168

Generated after EasyDMA has completed its 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	END			Generated after EasyDMA has completed its operation.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.12.2.2 EVENTS_DMA.TX.READY

Address offset: 0x16C

Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Generated when EasyDMA has buffered the .PTR and .MAXCNT registers for the channel, allowing them to be written to prepare for the next sequence.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.12.2.3 EVENTS_DMA.TX.BUSERROR

Address offset: 0x170

An error occured during the bus transfer.

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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	BUSERROR			An error occured during the bus transfer. When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.13 EVENTS_FRAMETIMEOUT

Address offset: 0x174

Timed out due to bus being idle while receiving 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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_FRAMETIMEOUT			Timed out due to bus being idle while receiving data.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

8.20.13.14 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event CTS will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.15 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event NCTS will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.16 PUBLISH_TXDRDY

Address offset: 0x18C

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event TXDRDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.17 PUBLISH_RXDRDY

Address offset: 0x190

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event RXDRDY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.18 PUBLISH_ERROR

Address offset: 0x194

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		[0..255]	DPPI channel that event ERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.19 PUBLISH_RXTO

Address offset: 0x1A4

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event RXTO will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.20 PUBLISH_TXSTOPPED

Address offset: 0x1B0

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		[0..255]	DPPI channel that event TXSTOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21 PUBLISH_DMA

Publish configuration for events

8.20.13.21.1 PUBLISH_DMA.RX

Publish configuration for events

8.20.13.21.1.1 PUBLISH_DMA.RX.END

Address offset: 0x1CC

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.1.2 PUBLISH_DMA.RX.READY

Address offset: 0x1D0

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.1.3 PUBLISH_DMA.RX.BUSERROR

Address offset: 0x1D4

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.1.4 PUBLISH_DMA.RX.MATCH[n] (n=0..3)

Address offset: 0x1D8 + (n × 0x4)

Publish configuration for event MATCH[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		[0..255]	DPPI channel that event MATCH[n] will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.2 PUBLISH_DMA.TX

Publish configuration for events

8.20.13.21.2.1 PUBLISH_DMA.TX.END

Address offset: 0x1E8

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		[0..255]	DPPI channel that event END will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.2.2 PUBLISH_DMA.TX.READY

Address offset: 0x1EC

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		[0..255]	DPPI channel that event READY will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.21.2.3 PUBLISH_DMA.TX.BUSERROR

Address offset: 0x1F0

Publish configuration for event BUSERROR

When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event BUSERROR will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.22 PUBLISH_FRAMETIMEOUT

Address offset: 0x1F4

Publish configuration for event FRAMETIMEOUT

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 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		[0..255]	DPPI channel that event FRAMETIMEOUT will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.20.13.23 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				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	DMA_RX_END_DMA_RX_START			Shortcut between event DMA.RX.END and task DMA.RX.START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	DMA_RX_END_DMA_RX_STOP			Shortcut between event DMA.RX.END and task DMA.RX.STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	DMA_TX_END_DMA_TX_STOP			Shortcut between event DMA.TX.END and task DMA.TX.STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D-G	RW	DMA_RX_MATCH[i]_DMA_RX_ENABLEMA +1)%4] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.ENABLEMATCH[(i+1)%4] Allows daisy-chaining match events.
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
H-K	RW	DMA_RX_MATCH[i]_DMA_RX_DISABLEMATCH[i] (i=0..3)			Shortcut between event DMA.RX.MATCH[n] and task DMA.RX.DISABLEMATCH[n]
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
L	RW	FRAMETIMEOUT_DMA_RX_STOP			Shortcut between event FRAMETIMEOUT and task DMA.RX.STOP
			Disabled	0	Disable shortcut

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

8.20.13.24 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			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	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	TXDRDY			Enable or disable interrupt for event TXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	RXDRDY			Enable or disable interrupt for event RXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	RXTO			Enable or disable interrupt for event RXTO
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	TXSTOPPED			Enable or disable interrupt for event TXSTOPPED
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	DMARXEND			Enable or disable interrupt for event DMARXEND
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	DMARXREADY			Enable or disable interrupt for event DMARXREADY
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	DMARXBUSERROR			Enable or disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable
K-N	RW	DMARXMATCH[i] (i=0..3)			Enable or disable interrupt for event DMARXMATCH[i]
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	DMATXEND			Enable or disable interrupt for event DMATXEND
			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				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
P	RW	DMATXREADY			Enable or disable interrupt for event DMATXREADY
			Disabled	0	Disable
			Enabled	1	Enable
Q	RW	DMATXBUSERROR			Enable or disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	FRAMETIMEOUT			Enable or disable interrupt for event FRAMETIMEOUT
			Disabled	0	Disable
			Enabled	1	Enable

8.20.13.25 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				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 W1S	CTS			Write '1' to enable interrupt for event CTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	NCTS			Write '1' to enable interrupt for event NCTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1S	TXDRDY			Write '1' to enable interrupt for event TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1S	RXDRDY			Write '1' to enable interrupt for event RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1S	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1S	RXTO			Write '1' to enable interrupt for event RXTO
			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			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
G	RW W1S	TXSTOPPED			Write '1' to enable interrupt for event TXSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1S	DMARXEND			Write '1' to enable interrupt for event DMARXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1S	DMARXREADY			Write '1' to enable interrupt for event DMARXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW W1S	DMARXBUSERROR			Write '1' to enable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K-N	RW W1S	DMARXMATCH[i] (i=0..3)			Write '1' to enable interrupt for event DMARXMATCH[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1S	DMATXEND			Write '1' to enable interrupt for event DMATXEND
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1S	DMATXREADY			Write '1' to enable interrupt for event DMATXREADY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW W1S	DMATXBUSERROR			Write '1' to enable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW W1S	FRAMETIMEOUT			Write '1' to enable interrupt for event FRAMETIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.20.13.26 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			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 W1C	CTS			Write '1' to disable interrupt for event CTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	NCTS			Write '1' to disable interrupt for event NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW W1C	TXDRDY			Write '1' to disable interrupt for event TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW W1C	RXDRDY			Write '1' to disable interrupt for event RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW W1C	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW W1C	RXTO			Write '1' to disable interrupt for event RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW W1C	TXSTOPPED			Write '1' to disable interrupt for event TXSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW W1C	DMARXEND			Write '1' to disable interrupt for event DMARXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW W1C	DMARXREADY			Write '1' to disable interrupt for event DMARXREADY
			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				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
J	RW W1C	DMARXBUSERROR			Write '1' to disable interrupt for event DMARXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K-N	RW W1C	DMARXMATCH[i] (i=0..3)			Write '1' to disable interrupt for event DMARXMATCH[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW W1C	DMATXEND			Write '1' to disable interrupt for event DMATXEND
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW W1C	DMATXREADY			Write '1' to disable interrupt for event DMATXREADY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW W1C	DMATXBUSERROR			Write '1' to disable interrupt for event DMATXBUSERROR When this event is generated, the address which caused the error can be read from the BUSERRORADDRESS register.
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW W1C	FRAMETIMEOUT			Write '1' to disable interrupt for event FRAMETIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.20.13.27 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 W1C	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

Bit number				31 30 29 28 27 26 25 24 23 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
			Present	1	Read: error present
B	RW W1C	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 W1C	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 W1C	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

8.20.13.28 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

8.20.13.29 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) when UARTE has 16 MHz peripheral clock frequency
			Baud2400	0x0009D000	2400 baud (actual rate: 2396) when UARTE has 16 MHz peripheral clock frequency
			Baud4800	0x0013B000	4800 baud (actual rate: 4808) when UARTE has 16 MHz peripheral clock frequency
			Baud9600	0x00275000	9600 baud (actual rate: 9598) when UARTE has 16 MHz peripheral clock frequency

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 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
			Baud14400	0x003AF000	14400 baud (actual rate: 14401) when UARTE has 16 MHz peripheral clock frequency
			Baud19200	0x004EA000	19200 baud (actual rate: 19208) when UARTE has 16 MHz peripheral clock frequency
			Baud28800	0x0075C000	28800 baud (actual rate: 28777) when UARTE has 16 MHz peripheral clock frequency
			Baud31250	0x00800000	31250 baud when UARTE has 16 MHz peripheral clock frequency
			Baud38400	0x009D0000	38400 baud (actual rate: 38369) when UARTE has 16 MHz peripheral clock frequency
			Baud56000	0x00E50000	56000 baud (actual rate: 55944) when UARTE has 16 MHz peripheral clock frequency
			Baud57600	0x00EB0000	57600 baud (actual rate: 57554) when UARTE has 16 MHz peripheral clock frequency
			Baud76800	0x013A9000	76800 baud (actual rate: 76923) when UARTE has 16 MHz peripheral clock frequency
			Baud115200	0x01D60000	115200 baud (actual rate: 115108) when UARTE has 16 MHz peripheral clock frequency
			Baud230400	0x03B00000	230400 baud (actual rate: 231884) when UARTE has 16 MHz peripheral clock frequency
			Baud250000	0x04000000	250000 baud when UARTE has 16 MHz peripheral clock frequency
			Baud460800	0x07400000	460800 baud (actual rate: 457143) when UARTE has 16 MHz peripheral clock frequency
			Baud921600	0x0F000000	921600 baud (actual rate: 941176) when UARTE has 16 MHz peripheral clock frequency
			Baud1M	0x10000000	1 megabaud when UARTE has 16 MHz peripheral clock frequency

8.20.13.30 CONFIG

Address offset: 0x56C

Configuration of parity, hardware flow control, framesize, and packet 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				G F E E E E D C B B B A
Reset 0x00001000				**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	RW	HWFC			Hardware flow control
			Disabled	0	Disabled
			Enabled	1	Enabled
B	RW	PARITY			Parity
			Excluded	0x0	Exclude parity bit
			Included	0x7	Include parity bit
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
E	RW	FRAMESIZE			Set the data frame size
			9bit	9	9 bit data frame size. 9th bit is treated as address bit.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E E E E D C B B B 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
			8bit	8	8 bit data frame size.
			7bit	7	7 bit data frame size.
			6bit	6	6 bit data frame size.
			5bit	5	5 bit data frame size.
			4bit	4	4 bit data frame size.
F	RW	ENDIAN			Select if data is trimmed from MSB or LSB end when the data frame size is less than 8.
			MSB	0	Data is trimmed from MSB end.
			LSB	1	Data is trimmed from LSB end.
G	RW	FRAMETIMEOUT			Enable packet timeout.
			DISABLED	0	Packet timeout is disabled.
			ENABLED	1	Packet timeout is enabled.
			Disabled	0	Packet timeout is disabled.
			Enabled	1	Packet timeout is enabled.

8.20.13.31 ADDRESS

Address offset: 0x574

Set the address of the UARTE for RX when used in 9 bit data frame 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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			Set address

8.20.13.32 FRAMETIMEOUT

Address offset: 0x578

Set the number of UARTE bits to count before triggering packet 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 A A A A A A A A A
Reset 0x00000010				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0**
ID	R/W	Field	Value ID	Value	Description
A	RW	COUNTERTOP			Number of UARTE bits before timeout.

8.20.13.33 PSEL.TXD

Address offset: 0x604

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.20.13.34 PSEL.CTS

Address offset: 0x608

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.20.13.35 PSEL.RXD

Address offset: 0x60C

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.20.13.36 PSEL.RTS

Address offset: 0x610

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 B 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..7]	Port number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

8.20.13.37 DMA.RX.PTR

Address offset: 0x704

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.20.13.38 DMA.RX.MAXCNT

Address offset: 0x708

Maximum number of bytes in channel 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 channel buffer

8.20.13.39 DMA.RX.AMOUNT

Address offset: 0x70C

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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 A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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.

8.20.13.40 DMA.RX.TERMINATEONBUSERROR

Address offset: 0x71C

Terminate the transaction if a BUSERROR event is 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	ENABLE			
			Disabled	0	Disable
			Enabled	1	Enable

8.20.13.41 DMA.RX.BUSERRORADDRESS

Address offset: 0x720

Address of transaction that generated the last BUSERROR 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 A A A A A A A A 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	ADDRESS			

8.20.13.42 DMA.RX.MATCH

Registers to control the behavior of the pattern matcher engine

8.20.13.42.1 DMA.RX.MATCH.CONFIG

Address offset: 0x724

Configure individual match events

Bit number				31 30 29 28 27 26 25 24 23 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-D	RW	ENABLE[i] (i=0..3)			Enable match filter i
			Disabled	0	Match filter disabled
			Enabled	1	Match filter enabled
E-H	RW	ONESHOT[i] (i=0..3)			Configure match filter i as one-shot or continous One-shot operation will disable the filter on a match, while Continous operation will keep it enabled until explicitly disabled.
			Continuous	0	Match filter stays enabled until disabled by task
			Oneshot	1	Match filter stays enabled until next data word is received

8.20.13.42.2 DMA.RX.MATCH.CANDIDATE[n] (n=0..3)

Address offset: 0x728 + (n × 0x4)

The data to look for - any match will trigger the MATCH[n] event, if enabled.

Note: This register can be updated while a transfer is in progress, but the new value will not take effect until either the DMA is restarted or the match event is generated. That makes it possible to write a new set of match words which will be searched for immediately after the event triggers.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	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	DATA			Data to look for

8.20.13.43 DMA.TX.PTR

Address offset: 0x73C

RAM buffer 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 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	PTR			RAM buffer start address for this EasyDMA channel. This address is a word aligned Data RAM address.

Note: See the memory chapter for details about which memories are available for EasyDMA.

8.20.13.44 DMA.TX.MAXCNT

Address offset: 0x740

Maximum number of bytes in channel 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 channel buffer

8.20.13.45 DMA.TX.AMOUNT

Address offset: 0x744

Number of bytes transferred in the last transaction, updated after the END event.

Also updated after each MATCH 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	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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.

8.20.13.46 DMA.TX.TERMINATEONBUSERROR

Address offset: 0x754

Terminate the transaction if a BUSERROR event is 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																															
A	RW	ENABLE																																	
			Disabled	0																															
			Enabled	1																															

8.20.13.47 DMA.TX.BUSERRORADDRESS

Address offset: 0x758

Address of transaction that generated the last BUSERROR 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	A	A	A	A	A	A	A	A	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	R	ADDRESS																																	

8.21 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.

The main features of WDT are:

- Generates watchdog reset
- Optional pause of WDT when the CPU is sleeping or when it is stopped by the debugger
- Optional generation of non-maskable interrupt (NMI)
- Runs off the low-frequency clock source (LFCLK)

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 54.

8.21.1 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.

8.21.2 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.

8.21.3 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 Reset behavior on page 90. After a reset, the watchdog configuration registers are available for configuration.

See RESET — Reset control on page 88 for more information about reset sources.

The TIMEOUT event will also generate NMI interrupt, when NMI interrupt is supported. See the the instance's configuration in Instantiation on page 141 to see if NMI is supported.

8.21.4 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 525 after the watchdog has stopped, to avoid runaway code triggering the STOP task.

8.21.5 Registers

Instances

Instance	Domain	Base address	Description
WDT30	GLOBAL	0x40108000	Watchdog timer WDT30

Configuration

Instance	Domain	Configuration
WDT30	GLOBAL	Supports non-maskable interrupts (NMI).

Register overview

Register	Offset	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

8.21.5.1 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

8.21.5.2 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

8.21.5.3 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		[0..255]	DPPI channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.21.5.4 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		[0..255]	DPPI channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

8.21.5.5 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

8.21.5.6 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

8.21.5.7 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event TIMEOUT will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.21.5.8 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	CHIDX		[0..255]	DPPI channel that event STOPPED will publish to
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

8.21.5.9 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 W1S	TIMEOUT			Write '1' to enable interrupt for event TIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.21.5.10 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 W1C	TIMEOUT			Write '1' to disable interrupt for event TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.21.5.11 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 W1S	TIMEOUT			Write '1' to enable interrupt for event TIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.21.5.12 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 W1C	TIMEOUT			Write '1' to disable interrupt for event TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1C	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

8.21.5.13 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

8.21.5.14 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							

8.21.5.15 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							

8.21.5.16 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							

8.21.5.17 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				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	RW	SLEEP			Configure WDT to either be paused, or kept running, while the CPU is sleeping
					Note: This feature will not affect power consumed by the WDT.
			Pause	0	Pause WDT while the CPU is sleeping
			Run	1	Keep WDT running while the CPU is sleeping
B	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
C	RW	STOPEN			Allow stopping WDT
			Disable	0	Do not allow stopping WDT
			Enable	1	Allow stopping WDT

8.21.5.18 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

8.21.5.19 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

9 Debug

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

The main features of the debug system are the following:

- Access port connection for Arm Cortex-M33
  - Eight breakpoints
  - Four watchpoint comparators
  - Access protection through APPROTECT and ERASEPROTECT
- Serial wire debug (SWD) interface, protocol version 2 with multidrop support
- Control-access port (CTRL-AP) that enables device control when other debug access ports (DAP) have been disabled by the access port protection

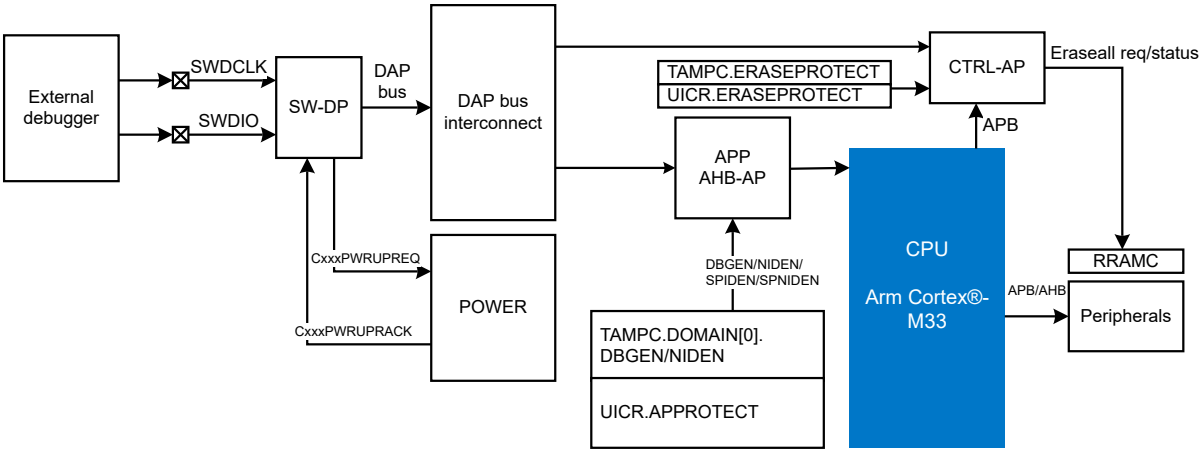


Figure 113: Debug overview

9.1 Debug access port

An external debugger can access the device through the debug access port (DAP).

The DAP implements a standard serial wire debug (SWD) Arm CoreSight protocol with a two-pin serial interface (**SWDCLK** and **SWDIO**).

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

The serial wire debug (SWD) pins can be used as GPIO pins when debugging is not active. For more information, see Pin assignments on page 546.

There are several access ports for connecting to different parts of the system, as shown in the following table.

AP ID	Type	Description
0	AHB-AP	CM33 access port
2	CTRL-AP	Control access port

Table 52: Access port overview

The AHB-AP is a standard Arm component. See the *ArmCoreSight SoC-400 Technical Reference Manual* for more information. The control access port (CTRL-AP) is proprietary and described in more detail in CTRL-AP — Control access port on page 531.

9.1.1 Debug access port registers

Register overview

Register	Offset	Description
TARGETID	0x024	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	0x034	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.

9.1.1.1 TARGETID

Address offset: 0x024

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		
Reset 0x00000289				**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**	**1**	**0**	**0**	**1**	
ID	R/W	Field	Value ID	Value																															
A	R	TDESIGNER																																	An 11-bit code JEDEC JEP106 continuation code and identity code. The ID identifies the designer of the part.
			NordicSemi	0x144																															Nordic Semiconductor ASA.
B	R	TPARTNO																																	Part number.
C	R	TREVISION																																	Target revision.

9.1.1.2 DLPIDR

Address offset: 0x034

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																																	Protocol version.
			SWDPv2	1																															SW protocol version 2.
B	R	TINSTANCE																																	Target instance.

9.2 Access port protection

The access ports can be protected to secure the internal assets and resources of the device. While the control access port (CTRL-AP) is always accessible from an external debugger, the system applies various protection mechanisms to control and restrict access to the individual AHB access ports.

Protection is controlled by specific registers, which enable or disable debug access at different levels. These registers are part of UICR and TAMPC. The access port is normally protected. The hardware and software configurations of these registers control the access protection policies as shown in the following table.

Registers	Description
UICR.APPROTECT	Hardware control of debug access. A device reset is required for this configuration to take effect. `Unprotected` – CPU controls DBGEN/NIDEN, locks disabled. Other values – DBGEN/NIDEN disabled and locked.
TAMPC.PROTECT.DOMAIN[0].DBGEN TAMPC.PROTECT.DOMAIN[0].NIDEN	CPU control of debug access. The registers can be locked.

Table 53: Arm Cortex-M33 AHB-AP debug access

The access port protection is illustrated in the following figures.

Figure 114: AUX access port protection overview

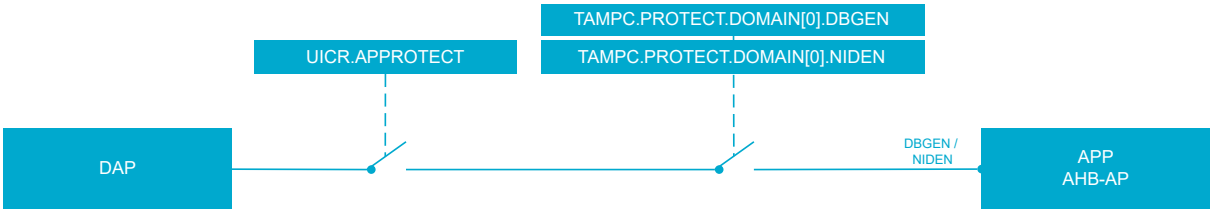


Figure 115: Arm Cortex-M33 AHB-AP access port protection overview

Registers	Description
UICR.APPROTECT	Hardware control of RRAMC ERASEALL protection in addition to access ports. A device reset is required for this configuration to take effect. Any value other than `Unprotected` disables RRAMC ERASEALL.
UICR.ERASEPROTECT	Hardware control of RRAMC ERASEALL and CTRL-AP ERASEALL protection. A device reset is required for this configuration to take effect. Any value other than `Unprotected` disables the erase all operations.
TAMPC.PROTECT.ERASEPROTECT	Software control of RRAMC ERASEALL and CTRL-AP ERASEALL protection. The register can be locked.

Table 54: Erase protection

The ERASEALL protection is illustrated in the following figure:

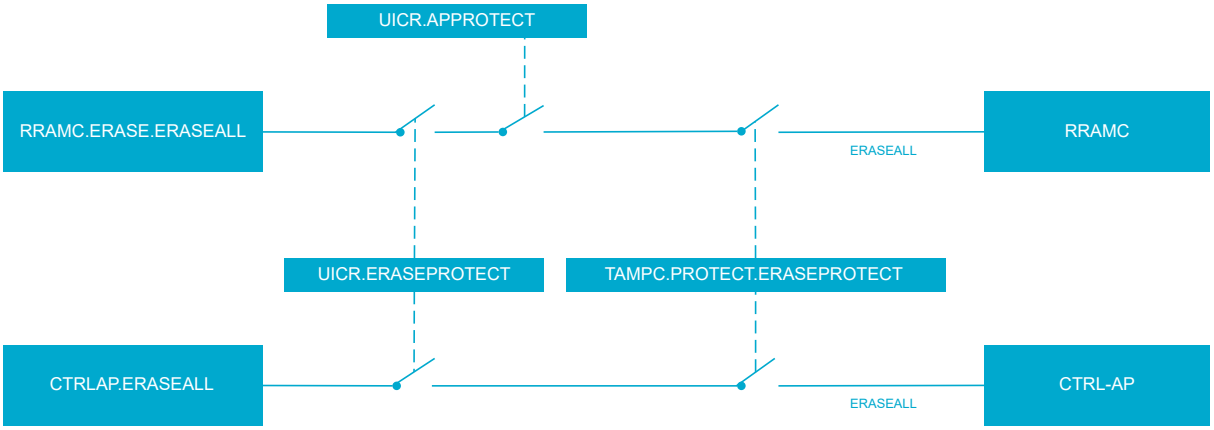


Figure 116: ERASEALL protection overview

The reset behavior of the TAMPC access port and ERASEALL protection is defined in Signal protector on page 125. On-chip software must write to the TAMPC registers before a debug access port is opened.

The access port remains open after the completion of the CTRL-AP.ERASEALL operation. CTRL-AP temporarily removes the access port protection until certain conditions are met, after which the protection will be reinstated. The AHB-AP will be protected when one of the following conditions are met:

- Power-on reset
- Brownout reset
- Watchdog timer reset
- Pin reset

The following figure shows how a device with access port protection enabled can be erased, programmed, and configured to allow debugging. The access port state is determined by operations sent from the debugger and registers written by firmware. Reset in the following figure refers to any of the conditions previously listed for AHB-AP protection. When writing to the TAMPC, the software must first disable write protection, then write the new values. For more details, see TAMPC — Tamper controller on page 123.

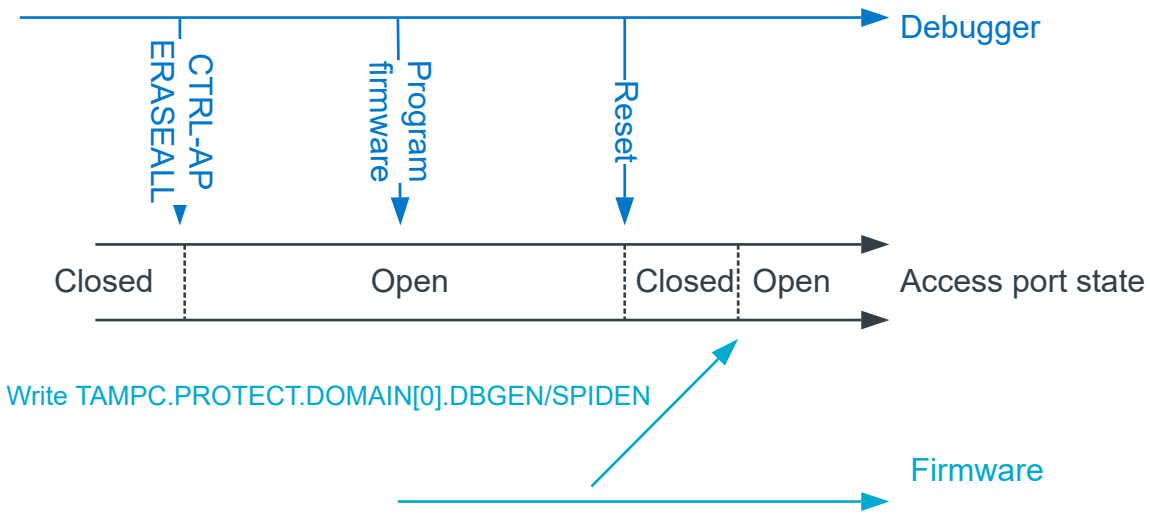


Figure 117: Access port unlocking

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*. 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 Debug access port on page 526.

9.3 Debug Interface mode

Before the external debugger can connect to an access port, the debugger must first request the device to power up through CxxxPWRUPREQ in the SWJ-DP.

The device remains in Debug Interface mode when the debugger requests power through CxxxPWRUPREQ. Otherwise, the device is in normal mode. When a debug session is over, the device must be set to normal mode by the external debugger, followed by a pin reset. This reduces overall power consumption.

Some peripherals behave differently in Debug Interface mode compared to normal mode. These differences are described in more detail in the corresponding peripheral chapter.

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 from CxxxPWRUPREQ, the system wakes up and the DIF flag in RESETREAS on page 91 is set.

9.4 Real-time debug

The device supports real-time debugging. This allows interrupts to execute to completion in real time when breakpoints are set in Thread mode or lower priority interrupts.

Real-time debugging enables setting a breakpoint for single-stepping through code. This prevents real-time event-driven threads from running at a higher priority. For example, this enables the device to continue to service 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.

9.5 Multidrop serial wire debug

Multidrop serial wire debug (SWD) allows simultaneous access to an unlimited number of devices through a single connection. This is useful for connectivity-constrained products that have several chips with multidrop support.

To select a target in a multidrop capable product, the debugger must write the correct TPARTNO, and TDESIGNER fields into the SW-DP TARGETSEL register. Values for these fields are located in DLPIDR on page 527.

For more information about multidrop SWD, see the *Arm Debug Interface Architecture Specification*, ADIv5.0 to ADIv5.2.

9.6 CTRL-AP — Control access port

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

For an overview of the other debug access ports, see DAP.

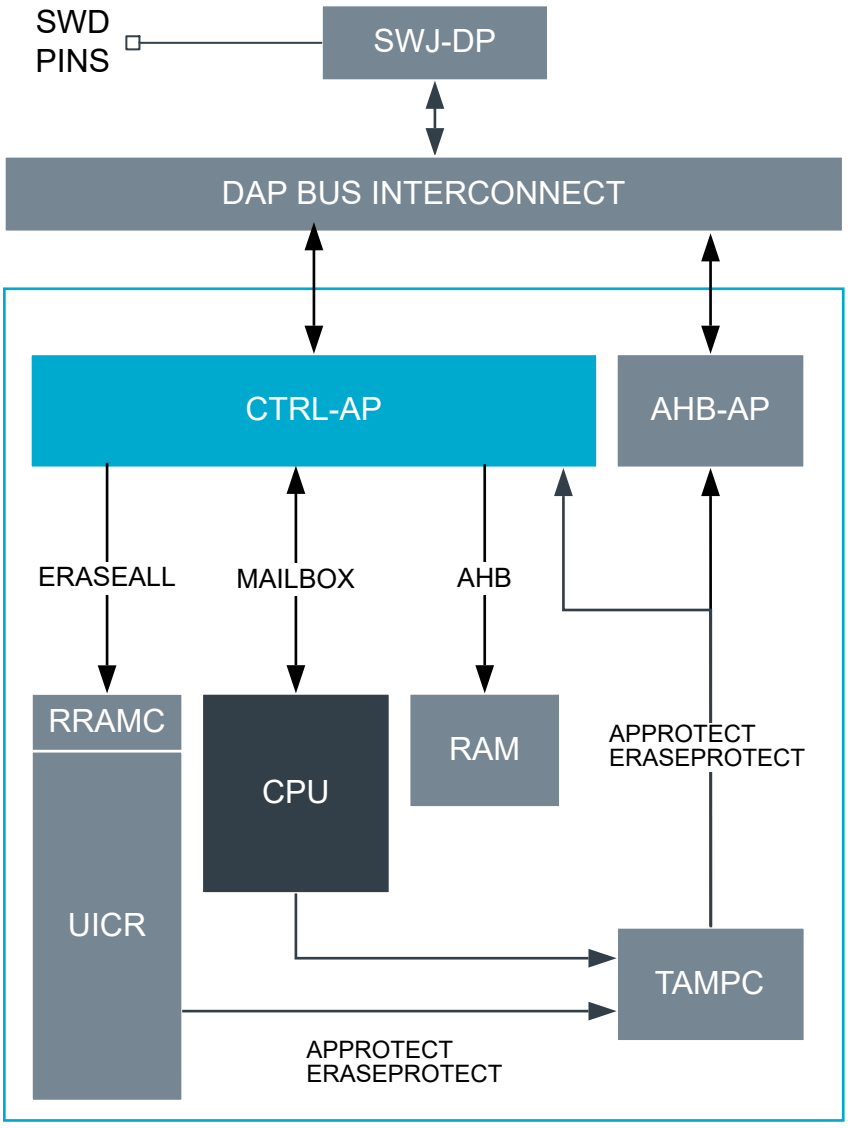


Figure 118: Control access port details

Access port protection (APPROTECT) blocks the debugger from accessing the AHB-AP and prevents read and write access to all CPU registers and memory-mapped addresses.

Erase protection (ERASEPROTECT) protects the entire non-volatile memory, and UICR from being erased. Erase protection is enabled through UICR and disabled through cooperation between the firmware and debugger. For more information about disabling erase protection, see Erase protection on page 534.

CTRL-AP has the following features:

- Reset the device
- Erase all
- Mailbox interface
- INFO.PARTNO and INFO.HWREVISION registers

The CTRL-AP peripheral has the following types of registers:

- Peripheral registers – Accessed by a CPU using the APB interface
- Debugger registers – Accessed by an external debugger using the CTRL-AP interface

9.6.1 Reset request

The debugger can request the device to perform a reset.

The register RESET is used to request the reset. Once the reset is performed, the reset reason is accessible through the RESETREAS register. For more information about reset, see RESET — Reset control on page 88.

9.6.2 Erase all

The erase all function lets the debugger trigger an erase of non-volatile memory, user information configuration registers (UICR), RAM, all peripheral settings, and also temporarily removes the access port protection.

To trigger an erase all function, follow the sequence in the following flowchart. After the sequence has completed, the access port protection is removed until the next pin reset, power-on reset, brownout reset, or watchdog timer reset.

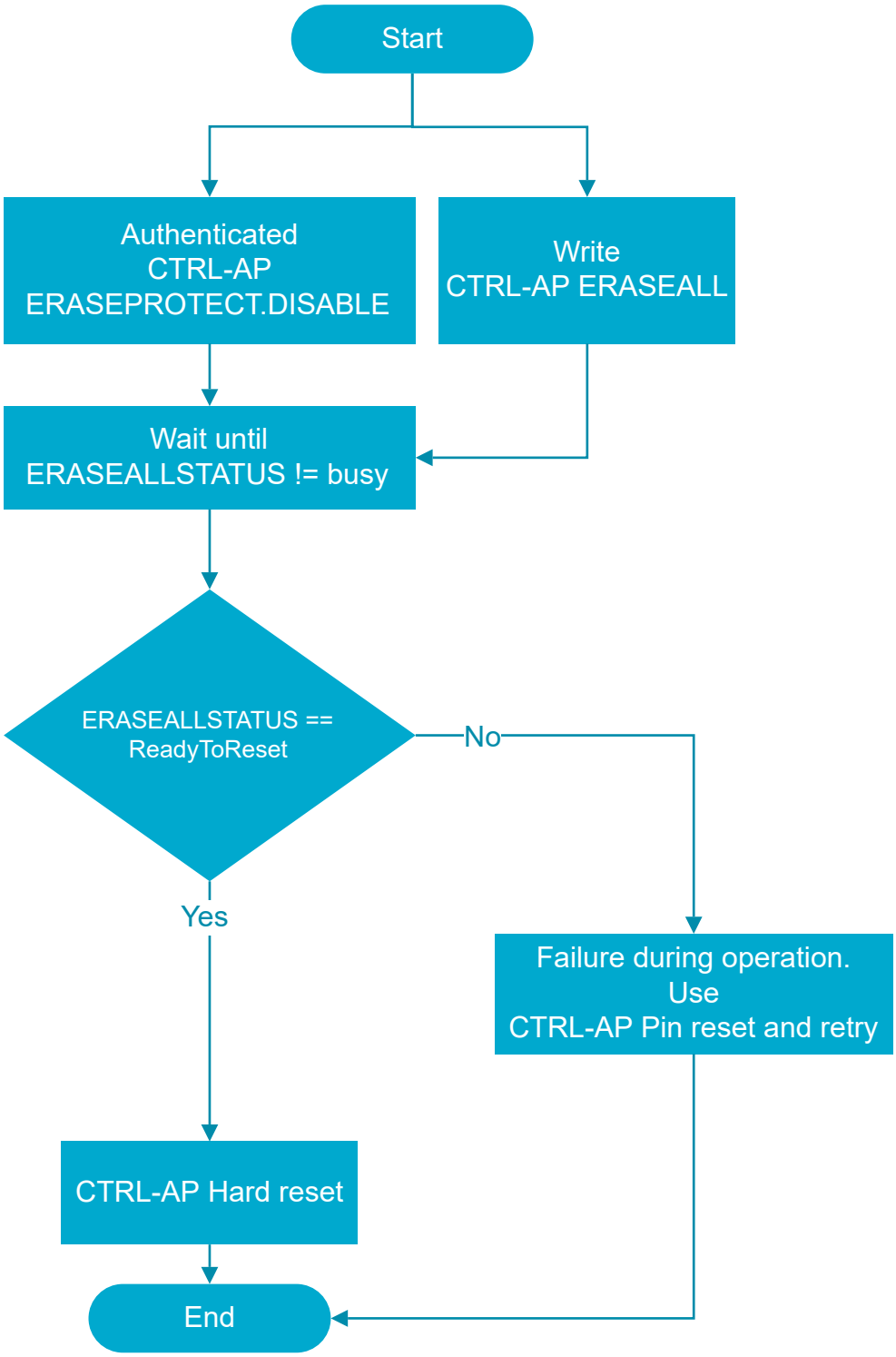


Figure 119: Erase all states

Erase all protection

It is possible to prevent the debugger from performing an erase all operation by using either the TAMPC TAMPC.PROTECT.ERASEPROTECT register, or the UICR.ERASEPROTECT register followed by a device reset.

Once this has been configured, the CTRL-AP ERASEALL operation is disabled.

The register ERASEPROTECT.STATUS on page 537 holds the status for erase protection.

9.6.2.1 Erase protection

Erase protection can be used to prevent a device from being erased.

A device ERASEALL operation can be initiated either by the non-volatile memory controller, or through the control access port. The following table describes the protection of the CTRL-AP ERASEALL operation. For more information, see Access port protection on page 528

Access port protection state		ERASEALL operation
UICR.ERASEPROTECT	TAMPC.PROTECT.ERASEPROTECT	ERASEALL
Unprotected	Unprotected	Allowed
Protected	-	Disabled
-	Protected	Disabled

Table 55: Erase all protection

The debugger can read the erase protection status in the register ERASEPROTECT.STATUS on page 537.

When erase protection is enabled, both the debugger and on-board firmware are required to disable it. The same non-zero 32-bit KEY value must be written to the debugger register ERASEPROTECT.DISABLE and CPU register ERASEPROTECT.DISABLE to disable 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 532. The access ports will be re-enabled on the next reset once the secure erase sequence has completed.

Write to the write-once register ERASEPROTECT.LOCK on page 543 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 pin reset, power-on reset, brownout reset, or watchdog timer reset, and therefore ERASEPROTECT.DISABLE is also disabled.

9.6.3 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 538 with its corresponding status register MAILBOX.TXSTATUS on page 538, and a receive register MAILBOX.RXDATA on page 538 with its corresponding status register MAILBOX.RXSTATUS on page 538. Status bits in registers TXSTATUS/RXSTATUS will be set and cleared automatically when registers TXDATA/RXDATA are written to and read from, independently of the direction.

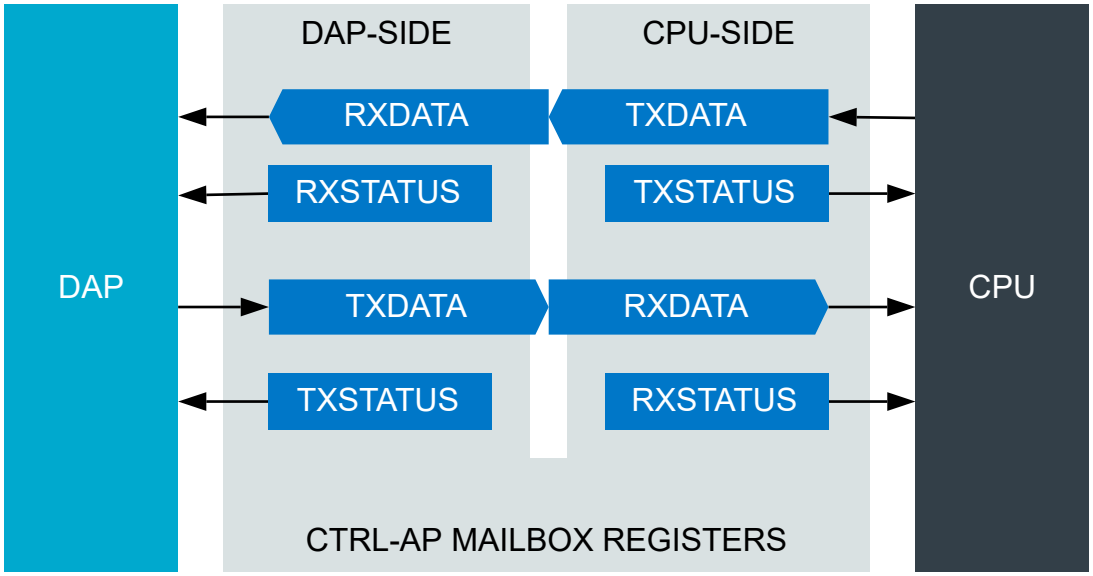


Figure 120: Mailbox register interface

Mailbox transfer sequence

1. Sender writes TXDATA
2. CTRL-AP sets sender's TXSTATUS to `DataPending`
3. CTRL-AP sets receiver's RXSTATUS to `DataPending`
4. Receiver reads RXDATA
5. CTRL-AP sets receiver's RXSTATUS to `NoDataPending`
6. CTRL-AP sets sender's TXSTATUS to `NoDataPending`

Events

EVENTS_RXREADY is generated when MAILBOX.RXSTATUS changes to `DataPending`. This indicates that a debugger has written new data to MAILBOX.RXDATA.

EVENTS_TXDONE is generated when the MAILBOX.TXSTATUSchanges to to `NoDataPending`. This indicates that a debugger has read the data from MAILBOX.TXDATA.

9.6.4 Device information

Device information, such as part number and hardware revision, can be read using CTRL-AP.

CTRL-AP provides the following information about the device:

- CTRL-AP identification register, IDR, see IDR
- Part number, see INFO.PARTNO
- Hardware revision, see INFO.HWREVISION

The information is available even for protected devices.

9.6.5 Debugger registers

CTRL-AP has a set of registers that can only be accessed from the debugger through the SWD interface. These are not accessible from the CPU.

9.6.5.1 Debug side registers

Register overview

Register	Offset	Description
RESET	0x000	System reset request and status
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.
ERASEPROTECT.STATUS	0x00C	Erase protection status.
ERASEPROTECT.DISABLE	0x010	This register disables ERASEPROTECT and performs Erase all.
APPROTECT.STATUS	0x014	This is the status register for the access port protection.
MAILBOX.TXDATA	0x020	Data sent from the debugger to the device.
MAILBOX.TXSTATUS	0x024	Status to indicate if data sent from the debugger to the device has been read.
MAILBOX.RXDATA	0x028	Data sent from the device to the debugger.
MAILBOX.RXSTATUS	0x02C	Status to indicate if data sent from the device to the debugger has been read.
INFO.PARTNO	0x030	Part number of the device
INFO.HWREVISION	0x034	Hardware Revision of the device
IDR	0x0FC	CTRL-AP Identification Register, IDR

9.6.5.1.1 RESET

Address offset: 0x000

System reset request and status

Only the enumerated values are supported, writing other values has unpredictable effect.

The Erase all operation cannot be interrupted by this reset request. Once the Erase all operation has started, CTRL-AP reset requests are 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																																	A	A	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	W	RESET																																	
			NoReset	0																															
			SoftReset	1																															
			HardReset	2																															
			PinReset	4																															

9.6.5.1.2 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**
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

9.6.5.1.3 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 A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 bits for the ERASEALL operation
			Ready	0	ERASEALL is ready
			ReadyToReset	1	Device is ready to be reset
			Busy	2	ERASEALL is busy (on-going)
			Error	3	Error during ERASEALL

9.6.5.1.4 ERASEPROTECT.STATUS

Address offset: 0x00C

Erase protection 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**
ID	R/W	Field	Value ID	Value	Description
A	R	PALL			Erase protection status.
			Enabled	1	Erase protection is enabled.
			Disabled	0	Erase protection is not enabled and ERASEALL can be performed.

9.6.5.1.5 ERASEPROTECT.DISABLE

Address offset: 0x010

This register disables ERASEPROTECT and performs Erase all.

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

9.6.5.1.6 APPROTECT.STATUS

Address offset: 0x014

This is the status register for the access port protection.

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

9.6.5.1.7 MAILBOX.TXDATA

Address offset: 0x020

Data sent from the debugger to the device.

Writing to this register will automatically set field DataPending in register TXSTATUS.

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

9.6.5.1.8 MAILBOX.TXSTATUS

Address offset: 0x024

Status to indicate if data sent from the debugger to the device 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																															
A	R	Status																																	
			NoDataPending	0																															
			DataPending	1																															

9.6.5.1.9 MAILBOX.RXDATA

Address offset: 0x028

Data sent from the device to the debugger.

Reading from this register will automatically set field NoDataPending in register RXSTATUS.

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

9.6.5.1.10 MAILBOX.RXSTATUS

Address offset: 0x02C

Status to indicate if data sent from the device 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																															
A	R	Status																																	
			NoDataPending	0																															
			DataPending	1																															

A	R	Status	Status of register DATA
		NoDataPending	No data pending in register RXDATA
		DataPending	Data pending in register RXDATA

9.6.5.1.11 INFO.PARTNO

Address offset: 0x030

Part number of the device

This register is retained on system on 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	A	A	A	A	A	A	A	A	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	R	PARTNO																																	

A	R	PARTNO	Part number

9.6.5.1.12 INFO.HWREVISION

Address offset: 0x034

Hardware Revision of the device

This register is retained on system on 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	A	A	A	A	A	A	A	A	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	R	ID																																	

A	R	ID	Part Variant

9.6.5.1.13 IDR

Address offset: 0x0FC

CTRL-AP Identification Register, IDR

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				E	E	E	E	D	D	D	D	C	C	C	C	C	C	C	B	B	B	B						A	A	A	A	A	A	A	A
Reset 0x32880000				**0**	**0**	**1**	**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																															
A	R	APID																																	
B	R	CLASS																																	
			NotDefined	0x0																															
			MEMAP	0x8																															
C	R	JEP106ID																																	
D	R	JEP106CONT																																	
E	R	REVISION																																	

A	R	APID	AP Identification
B	R	CLASS	Access Port (AP) class
		NotDefined	No defined class
		MEMAP	Memory Access Port
C	R	JEP106ID	JEDEC JEP106 identity code
D	R	JEP106CONT	JEDEC JEP106 continuation code
E	R	REVISION	Revision

9.6.6 Peripheral registers

CTRL-AP has a set of registers that can be accessed by the CPU.

9.6.6.1 CPU side registers

Instances

Instance	Domain	Base address	Description
CTRLAP	GLOBAL	0x40052000	Control access port CPU side

Register overview

Register	Offset	Description
EVENTS_RXREADY	0x100	RXSTATUS is changed to DataPending.
EVENTS_TXDONE	0x104	TXSTATUS is changed to NoDataPending.
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
INTPEND	0x30C	Pending interrupts
MAILBOX.RXDATA	0x400	Data sent from the debugger to the CPU.
MAILBOX.RXSTATUS	0x404	Status to indicate 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	Status to indicate if 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.
RESET	0x520	System reset request.

9.6.6.1.1 EVENTS_RXREADY

Address offset: 0x100

RXSTATUS is changed to DataPending.

New data is available in MAILBOX.RXDATA.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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			RXSTATUS is changed to DataPending. New data is available in MAILBOX.RXDATA.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

9.6.6.1.2 EVENTS_TXDONE

Address offset: 0x104

TXSTATUS is changed to NoDataPending.

MAILBOX.TXDATA 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	RW	EVENTS_TXDONE			TXSTATUS is changed to NoDataPending. MAILBOX.TXDATA has been read.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

9.6.6.1.3 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				B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	R/W	Field	Value ID	Value	Description
A	RW	RXREADY			Enable or disable interrupt for event RXREADY New data is available in MAILBOX.RXDATA.
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	TXDONE			Enable or disable interrupt for event TXDONE MAILBOX.TXDATA has been read.
			Disabled	0	Disable
			Enabled	1	Enable

9.6.6.1.4 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 W1S	RXREADY			Write '1' to enable interrupt for event RXREADY New data is available in MAILBOX.RXDATA.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW W1S	TXDONE			Write '1' to enable interrupt for event TXDONE MAILBOX.TXDATA has been read.
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

9.6.6.1.5 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																															
A	RW W1C	RXREADY																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															
B	RW W1C	TXDONE																																	
			Clear	1																															
			Disabled	0																															
			Enabled	1																															

9.6.6.1.6 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																																		B	A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	R	RXREADY																																	
			NotPending	0																															
			Pending	1																															
B	R	TXDONE																																	
			NotPending	0																															
			Pending	1																															

9.6.6.1.7 MAILBOX.RXDATA

Address offset: 0x400

Data sent from the debugger to the CPU.

Reading from this register will automatically set field NoDataPending in register RXSTATUS.

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

9.6.6.1.8 MAILBOX.RXSTATUS

Address offset: 0x404

Status to indicate 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 is pending in register RXDATA.
			DataPending	1	Data is pending in register RXDATA.

9.6.6.1.9 MAILBOX.TXDATA

Address offset: 0x480

Data sent from the CPU to the debugger.

Writing to this register will automatically set field DataPending in register TXSTATUS.

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

9.6.6.1.10 MAILBOX.TXSTATUS

Address offset: 0x484

Status to indicate 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	TXSTATUS			Status of data in register TXDATA.
			NoDataPending	0	No data is pending in register TXDATA.
			DataPending	1	Data is pending in register TXDATA.

9.6.6.1.11 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			Writing any value will lock the ERASEPROTECT.DISABLE register from being written until next reset.
			Locked	1	Register ERASEPROTECT.DISABLE is read-only.

9.6.6.1.12 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	W1	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.

9.6.6.1.13 RESET

Address offset: 0x520

System reset request.

Only the enumerated values are supported, writing other values has unpredictable 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	A	A
Reset 0x00000000	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	RESET			Reset request
			NoReset	0	No reset is generated
			SoftReset	1	Perform a device soft reset
			HardReset	2	Perform a device hard reset
			PinReset	4	Perform a device pin reset

9.7 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.

9.7.1 Registers

Instances

Instance	Domain	Base address	Description
TAD	GLOBAL	0x40053000	Trace and debug control

Configuration

Instance	Domain	Configuration
TAD	GLOBAL	Has two TPIUs.

Register overview

Register	Offset	Description
SYSPWRUPREQ	0x400	System power-up request
DBGPWRUPREQ	0x404	Debug power-up request

9.7.1.1 SYSPWRUPREQ

Address offset: 0x400

System power-up request

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																			A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	ACTIVE																																	Activate power-up request
			NotActive	0																															Power-up request not active
			Active	1																															Power-up request active

9.7.1.2 DBGPWRUPREQ

Address offset: 0x404

Debug power-up request

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																			A
Reset 0x00000000				**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**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	ACTIVE																																	Activate power-up request
			NotActive	0																															Power-up request not active
			Active	1																															Power-up request active

10 Hardware and layout

10.1 Pin assignments

The pin assignment figures and tables describe the pinouts for the product variants of the device.

The nRF54LS05A/nRF54LS05B device provides flexibility regarding GPIO pin routing and configuration. Some pins have limitations or recommendations on how the pin should be configured or what it should be used for.

As a general rule, peripherals must use GPIO pins in their own domain, and can be used for all peripheral functions when selected in the peripheral's PSEL register. Note the dedicated clock pin requirements listed in the following chapter. In addition, there are some dedicated pin functions that allows pin connections between different power domains.

The block diagram shows which peripheral and port belong together, see Block diagram on page 8.

The different GPIO ports have different properties. For details, see GPIO — General purpose input/output on page 182.

10.1.1 Dedicated pins

The device has some pins dedicated for specific purposes. GPIO pin routing and configuration is flexible. Some pins have limitations or recommendations for configuration and use.

Function or peripheral	Description
GRTC	Has dedicated pins for clock and PWM output.
SWD	The debug interface has dedicated pins. With some restrictons, these pins can also be used as GPIO pins as described in the following section.
Weak pull-ups	A set of GPIO pins have additional pull-up options, all enabled by a single registers. For more details, see GPIO — General purpose input/output on page 182.

Table 56: Dedicated pin functions

10.1.2 SWD pins

The serial wire debug (SWD) pins can be used as GPIO pins when debugging is not active.

These pins default to SWD functionality at reset, and can be configured as GPIO pins using the PROTECT.GPIOSWD signal protector interface of TAMPC. Once configured as GPIO pins, the SWD functionality and all SWD-related debug operations are disabled until the next reset. The debugger will not be able to wake the device from sleep or from System OFF mode through the SWD interface, but must instead use the pin reset (**nRESET**) to reset the device.

The pins shall only be used as outputs, connected to high impedant loads. This ensures that the pins do not load the debug interface when debugging is enabled. Failure to do so can cause the device SWD operations to malfunction.

The debugger can prevent the device from using the SWD pins as GPIOs by holding **SWDCLK** high when **nRESET** is de-asserted. This forces the device to keep the SWD functionality enabled.

10.1.3 Clock pins

The device has dedicated clock pins.

Some peripherals have clock signals. Dedicated clock pins have been optimized to ensure correct timing relationship between clock and data signals for these peripherals. See the following table for which peripheral signals must use clock pins. The pin assignment table identifies clock pins.

Clock pins can also be used as regular I/O data pins.

The peripheral data signal must be configured to use pins close to the clock pin. This ensures that the internal paths from the peripheral to the pin have the same delay, so that the data and clock signals reach the pins at the same time.

For high-speed signals, the printed circuit board (PCB) layout must use short PCB traces of identical length. This makes sure any delays are kept to a minimum, with close to identical delay on the clock and data path.

The following table shows which peripheral signals must use clock pins.

Peripheral	Signal	Clock pin required
SPIM/SPIS	SDO	
	SDI	
	SCK	x
	CSN	
	DCX	
TWIM/TWIS	SCL	x
GRTC	CLKOUT32K	x (dedicated pin)
	PWM	x (dedicated pin)
	CLKOUTFAST	x (dedicated pin)

Table 57: List of peripheral signals and clock pin requirement

10.1.4 QFN48 (QFAA) package pin assignments

The QFN48 pin assignment figure and table describe the pinouts for this variant of the device.

Pins that can be used as clock signals are shown in as red in the figure. For more information about clock pins, see Clock pins on page 547.

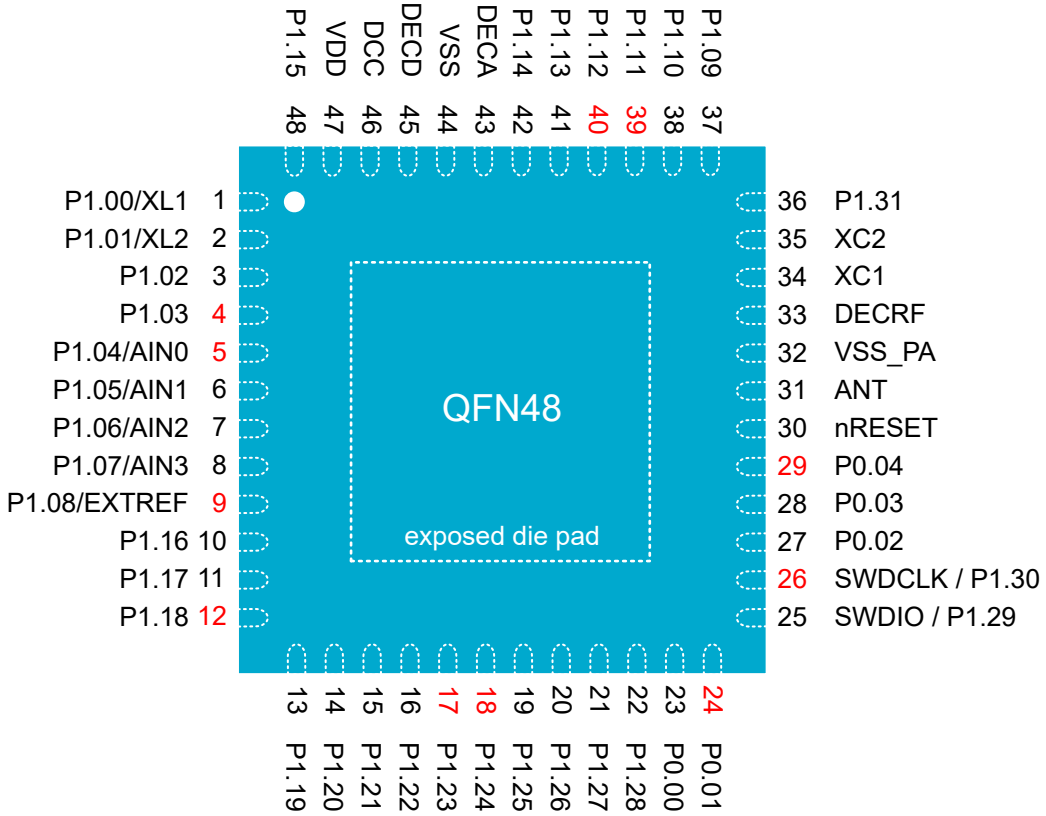


Figure 121: QFN48 pin assignments, top view

Pin	Clock pin	Weak pull-up	Name	Function	Description	Dedicated function
1			P1.00 XL1	Digital I/O Analog input	General purpose I/O Connection for 32.768 kHz crystal	
2			P1.01 XL2	Digital I/O Analog input	General purpose I/O Connection for 32.768 kHz crystal	
3			P1.02	Digital I/O	General purpose I/O	
4	Yes		P1.03	Digital I/O	General purpose I/O	
5	Yes		P1.04 AIN0	Digital I/O Analog input	General purpose I/O Analog input	
6			P1.05 AIN1	Digital I/O Analog input	General purpose I/O Analog input	
7			P1.06 AIN2	Digital I/O Analog input	General purpose I/O Analog input	
8			P1.07 AIN3	Digital I/O Analog input	General purpose I/O Analog input	
9	Yes		P1.08 EXTREF	Digital I/O Digital I/O Analog input	General purpose I/O GRTC CLKOUTFAST External reference for SAADC	GRTC
10			P1.16	Digital I/O	General purpose I/O	
11		Yes	P1.17	Digital I/O	General purpose I/O	
12	Yes	Yes	P1.18	Digital I/O	General purpose I/O	
13	Yes	Yes	P1.19	Digital I/O	General purpose I/O	
14		Yes	P1.20	Digital I/O	General purpose I/O	
15		Yes	P1.21	Digital I/O	General purpose I/O	
16		Yes	P1.22	Digital I/O	General purpose I/O	
17	Yes	Yes	P1.23	Digital I/O	General purpose I/O	
18	Yes	Yes	P1.24	Digital I/O	General purpose I/O	

Pin	Clock pin	Weak pull-up	Name	Function	Description	Dedicated function
19		Yes	P1.25	Digital I/O	General purpose I/O	
20		Yes	P1.26	Digital I/O	General purpose I/O	
21		Yes	P1.27	Digital I/O Weak pull-up	General purpose I/O Weak pull-up	
22			P1.28	Digital I/O	General purpose I/O	
23			P0.00	Digital I/O	General purpose I/O	
24	Yes		P0.01	Digital I/O Digital I/O	General purpose I/O GRTC PWM	GRTC
25			SWDIO P1.29	Debug General purpose I/O	Serial wire data. Bidirectional with standard-drive and on-chip pull-down. GPIO with limitations.	See SWD pins on page 546.
26	Yes		SWDCLK P1.30	Debug General purpose I/O	Serial wire clock. Input with on-chip pull-up. GPIO with limitations.	See SWD pins on page 546.
27			P0.02	Digital I/O	General purpose I/O	
28			P0.03	Digital I/O	General purpose I/O	
29	Yes		P0.04	Digital I/O Digital I/O	General purpose I/O GRTC CLKOUT32K	GRTC
30			nRESET	Reset	Pin reset with on-chip pull-up	
31			ANT	RF	Single ended radio antenna connection	See Reference circuitry on page 552 for guidelines on how to ensure good RF performance.
32			VSS_PA	Power	Ground (radio supply)	
33			DECRF	Power	0.9 V regulator supply decoupling	Must be connected to DECA. See Reference

Pin	Clock pin	Weak pull-up	Name	Function	Description	Dedicated function
						circuitry on page 552.
34			XC1	Analog input	Connection for 32 MHz crystal	
35			XC2	Analog input	Connection for 32 MHz crystal	
36			P1.31	Digital I/O	General purpose I/O	
37			P1.09	Digital I/O	General purpose I/O	
38			P1.10	Digital I/O	General purpose I/O	
39	Yes		P1.11	Digital I/O	General purpose I/O	
40	Yes		P1.12	Digital I/O	General purpose I/O	
41			P1.13	Digital I/O	General purpose I/O	
42			P1.14	Digital I/O	General purpose I/O	
43			DECA	Power	0.9 V regulator supply decoupling	Must be connected to DECRF. See Reference circuitry on page 552.
44			VSS	Power	Ground	
45			DECD	Power	0.9 V regulator supply decoupling	
46			DCC	Power	DC/DC regulator output	
47			VDD	Power	Power supply	
48			P1.15	Digital I/O	General purpose I/O	
49			VSS	Power	Ground pad (die pad)	

Table 58: QFN48 pin assignments

10.2 Mechanical specifications

The mechanical specifications for the packages show the dimensions in millimeters.

10.2.1 QFN48 (QFAA) package

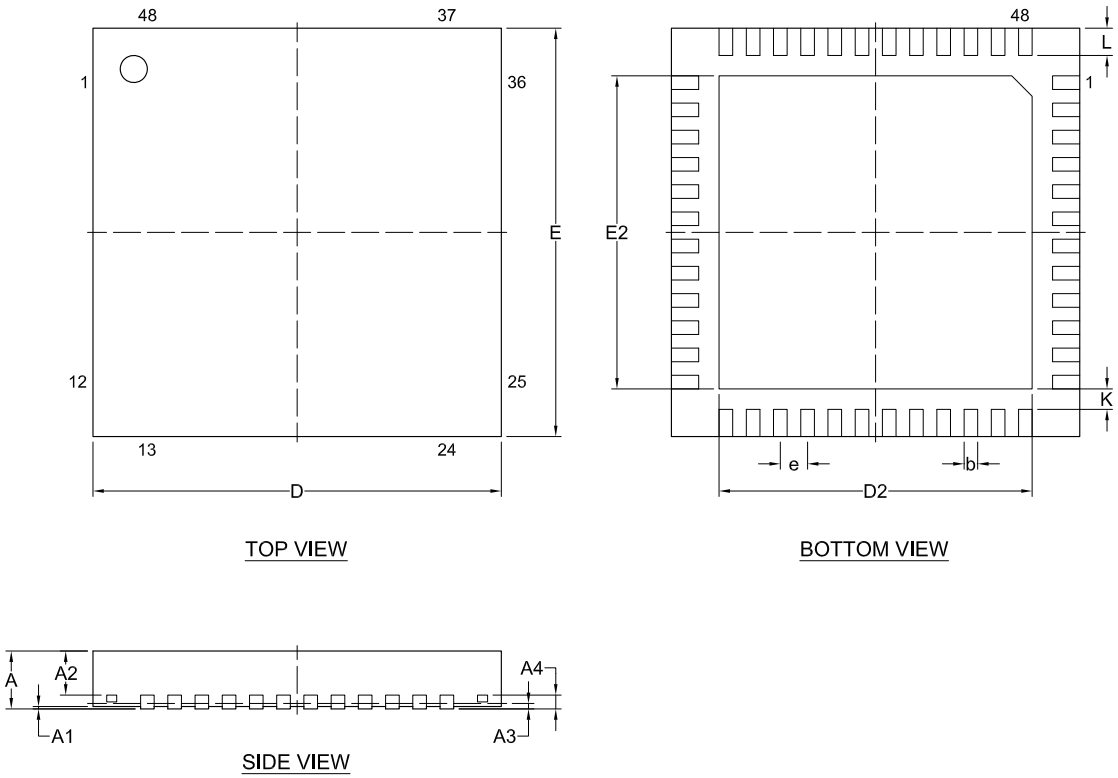


Figure 122: QFN48 6x6 mm package

	A	A1	A2	A3	A4	b	D, E	D2, E2	e	K	L
Min.	0.80	0.00	0.6		0.153	0.15	5.9	4.5		0.2	0.35
Nom.	0.85	0.04	0.65	0.8	0.203	0.20	6.0	4.6	0.40	0.3	0.40
Max.	0.90	0.05	0.70		0.253	0.25	6.1	4.7			0.45

Table 59: Package dimensions in millimeters

10.3 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.

The reference circuit is not included in the datasheet until validated. For access to preliminary design files, refer to the product page on nordicsemi.com.

10.4 Reflow conditions

The maximum amount of reflow is three, with each reflow at 260°C and 30 seconds.

11 Electrical specifications

11.1 Current consumption

Because the power and clock management system is constantly adjusting power and clock sources, it is difficult to estimate an application's current consumption. To facilitate the estimation process, a set of current consumption scenarios is provided to show the typical current drawn from the supply pins.

Each scenario specifies a set of operations and conditions applying to the given scenario. All scenarios are listed in the following sections.

11.1.1 CURRENT Electrical specification

11.1.1.1 Sleep

Symbol	Description	Min.	Typ.	Max.	Units
I_{OFF0}	System OFF, Wake on pin, 0 KB RAM retained		Estimated 0.6		µA
I_{OFF1}	System OFF, Wake on pin + GRTC, LFXO, 0 KB RAM retained		Estimated 0.7		µA
I_{ONIDLE}	System ON, Wake on pin + GRTC, LFXO, 96 KB RAM		Estimated 1.2		µA

11.1.1.2 CPU running

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

Compiler: Arm version 6.16 (armclang)

Compiler flags:

```
--target=arm-arm-none-eabi -c -g -masm=auto -Wno-unused-value -mcpu=cortex-m33 -mfloat-abi=hard -mfpu=fpv5-sp-d16 -flto -Omax
```

Linker flags:

```
--lto --remove
```

Symbol	Description	Min.	Typ.	Max.	Units
$I_{APPCPU0}$	CPU running Coremark at 128 MHz from NVM, Cache enabled		Estimated 2.0		mA

11.1.1.3 RADIO transmitting/receiving

Symbol	Description	Min.	Typ.	Max.	Units
I_{RADIO_RX0}	Radio RX, 1 Mbps, HFXO		Estimated 3.4		mA
I_{RADIO_TX0}	Radio TX, 0 dBm, HFXO		Estimated 4.8		mA
I_{RADIO_TX1}	Radio TX, 4 dBm, HFXO		Estimated 6.6		mA

11.2 CLOCK Electrical specification

11.2.1 High frequency clock source (HFCLK)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM}	Nominal output frequency		128		MHz

11.2.2 32.768 kHz clock source (LFCLK)

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

11.3 CPU Electrical specification

11.3.1 CPU performance

The CPU performance metrics are derived from benchmarks executed on highly optimized firmware. The firmware is compiled using the specified compiler version and settings to ensure accurate and reliable performance data.

Compiler: Arm version 6.16 (armclang)

Compiler flags:

```
--target=arm-arm-none-eabi -c -g -masm=auto -Wno-unused-value -mcpu=cortex-m33 -mfloat-abi=hard -mfpu=fpv5-sp-d16 -flto -Omax
```

Linker flags:

```
--lto --remove
```

Symbol	Description	Min.	Typ.	Max.	Units
$CM_{RRAMCACHE128}$	CPU running CoreMark at 128 MHz from RRAM, cache enabled		500		CoreMark

11.4 GPIO Electrical specification

11.4.1 GPIO Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage	0.7 x VDD		VDD	V
V_{IL}	Input low voltage	VSS		0.3 x VDD	V
$V_{OH,SD}$	Output high voltage, standard drive, 0.5 mA, VDD ≥1.7	VDD - 0.4		VDD	V
$V_{OH,HDH}$	Output high voltage, high drive, 5 mA, VDD >= 2.7 V	VDD - 0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 3 mA, VDD >= 1.7 V	VDD - 0.4		VDD	V
$V_{OL,SD}$	Output low voltage, standard drive, 0.5 mA, VDD ≥1.7	VSS		VSS + 0.4	V
$I_{OL,SD}$	Current at VSS+0.4 V, output set low, standard drive, VDD ≥1.7	1	3	4	mA
$I_{OL,HDL}$	Current at VSS+0.4 V, output set low, high drive, VDD >= 1.7 V	3			mA
$I_{OH,SD}$	Current at VDD-0.4 V, output set high, standard drive, VDD ≥1.7	1	3	4	mA
$I_{OH,HDL}$	Current at VDD-0.4 V, output set high, high drive, VDD >= 1.7 V	4			mA
$I_{GPIO,TOTAL}$	Recommended maximum sustained current drawn by all GPIOs			15	mA
R_{PU}	Pull-up resistance	12	14	16	kΩ
R_{PD}	Pull-down resistance	12	14	18	kΩ
C_{PAD}	Pad capacitance		1		pF

11.5 OSCILLATORS Electrical specification

11.5.1 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			±40	ppm

11.5.2 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

11.6 REGULATORS Electrical specification

11.6.1 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}	VDD supply voltage.	1.7		3.6	V

11.6.2 Power-fail comparator

Symbol	Description	Min.	Typ.	Max.	Units
V_{POF}	Voltage level warning thresholds (falling supply voltage). Levels are configurable between min. and max. in increments of 100 mV.	1.7		3.2	V
V_{POFTOL}	Threshold voltage tolerance.	-2		2	%
$V_{POFHYST}$	Threshold voltage hysteresis.	40	50	55	mV

11.7 RESET Electrical specification

11.7.1 Startup times

Symbol	Description	Min.	Typ.	Max.	Units
$t_{POR,10us}$	Time measured as time in power-on reset after supply reaches minimum operating voltage, with VDD rise time from 1 µs to 100ms.		0.2	2	ms

11.8 SPIM Electrical specification

11.8.1 Timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIM}	Bit rates for SPIM[2]			8	Mbps
$t_{SPIM,START}$	Time from START task to transmission started		1		µs
$t_{SPIM,CSCK}$	SCK period	31.25			ns
$t_{SPIM,RSCK,LD}$	SCK rise time, standard drive[3]			$t_{RF,25pF}$	
$t_{SPIM,RSCK,HD}$	SCK rise time, high drive[3]			$t_{HRF,25pF}$	
$t_{SPIM,FSCK,LD}$	SCK fall time, standard drive[3]			$t_{RF,25pF}$	
$t_{SPIM,FSCK,HD}$	SCK fall time, high drive[3]			$t_{HRF,25pF}$	
$t_{SPIM,WHSCK}$	SCK high time[3]	$(t_{CSCK}/2) - t_{RSCK} - 1.5$ ns			
$t_{SPIM,WLSCK}$	SCK low time[3]	$(t_{CSCK}/2) - t_{FSCK} - 1.5$ ns			

[2] High bit rates may require GPIOs to be set as High Drive or Extra High Drive, see GPIO chapter for more details.

[3] At 25pF load, including GPIO pin capacitance, see GPIO spec.

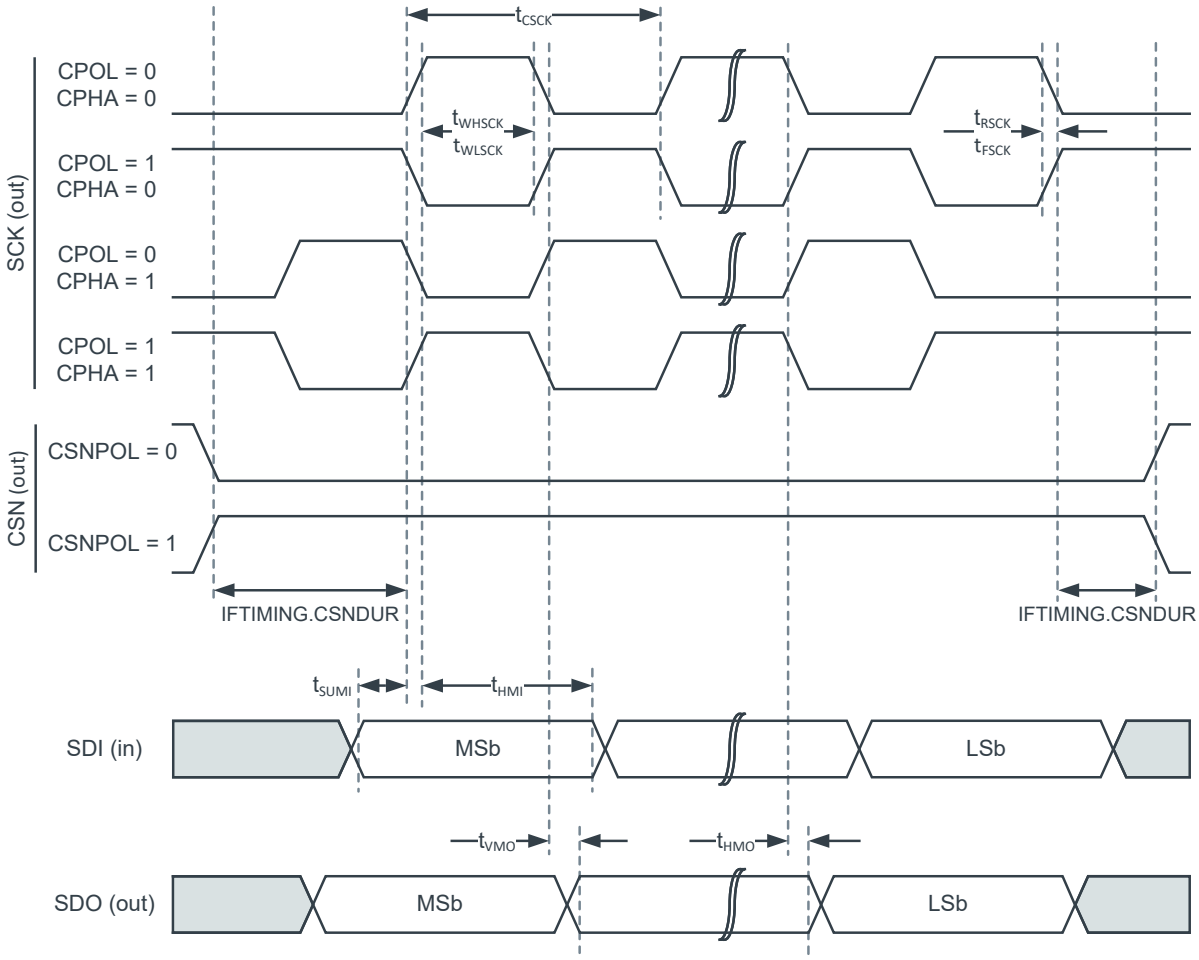


Figure 123: SPIM timing diagram

11.8.2 Timing specifications for GPIO port P1 using Standard drive strength

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,SUMI}$	SDI to CLK edge setup time	35			ns
$t_{SPIM,HMI}$	CLK edge to SDI hold time	0			ns
$t_{SPIM,VMO}$	CLK edge to SDO valid			4	ns
$t_{SPIM,HMO}$	SDO hold time after CLK edge	-4			ns

11.8.3 Timing specifications for GPIO port P1 using High drive strength

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,SUMI}$	SDI to CLK edge setup time	32			ns
$t_{SPIM,HMI}$	CLK edge to SDI hold time	0			ns
$t_{SPIM,VMO}$	CLK edge to SDO valid			7	ns
$t_{SPIM,HMO}$	SDO hold time after CLK edge	-6			ns

11.9 SPIS Electrical specification

11.9.1 SPIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIS}	Bit rates for SPIS[4]			8[5]	Mbps
$t_{SPIS,START}$	Time from RELEASE task to receive/transmit (CSN active)		1		µs

11.9.2 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[6]			ns
$t_{SPIS,HCSN}$	CLK to CSN hold time	1000			ns
$t_{SPIS,ASA}$	CSN to SDO driven			70	ns
$t_{SPIS,ASO}$	CSN to SDO valid[7]			1000	ns
$t_{SPIS,DISSO}$	CSN to SDO disabled[7]			70	ns
$t_{SPIS,CWH}$	CSN inactive time	300			ns

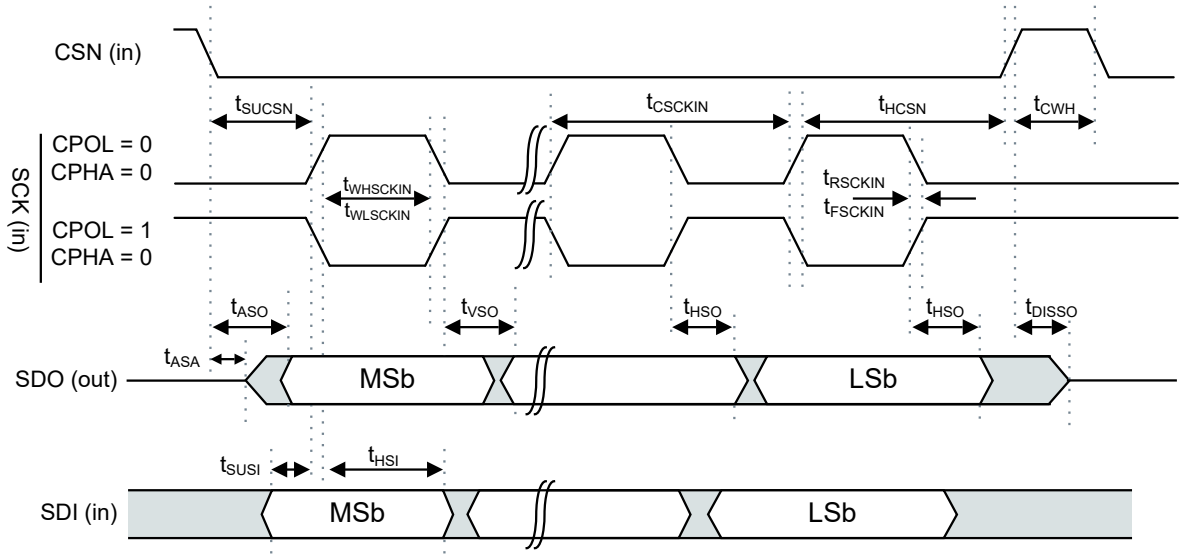


Figure 124: SPIS timing diagram, CPHA = 0

[4] High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.
[5] The actual maximum data rate depends on the master's CLK to SDO and SDI setup and hold timings.
[6] Excluding any start-up delay for the high frequency clock in low power mode.
[7] At 25pF load, including GPIO capacitance, see GPIO spec.

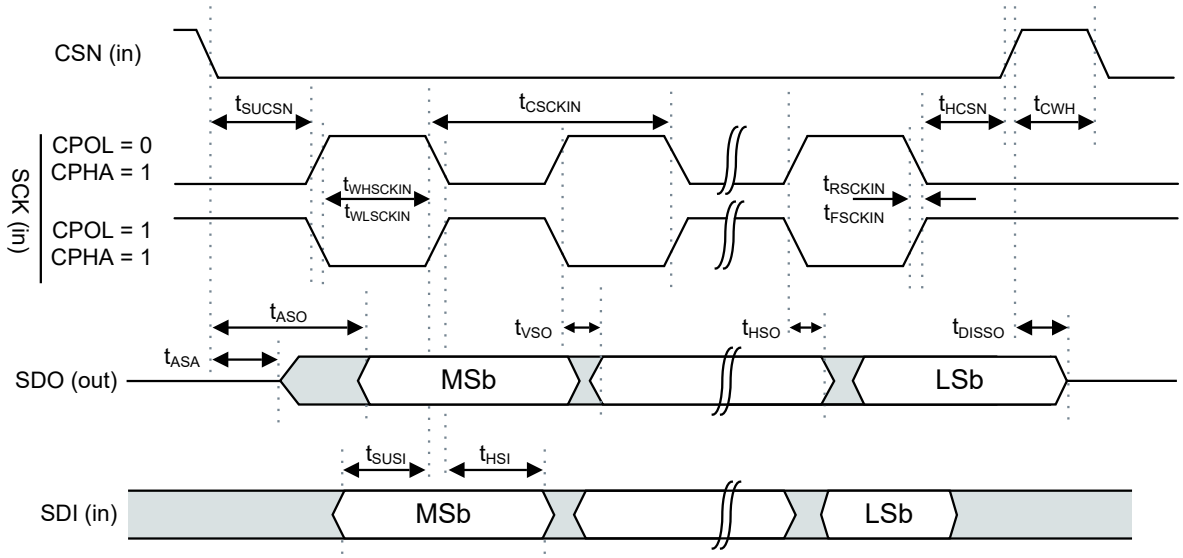


Figure 125: SPIS timing diagram, CPHA = 1

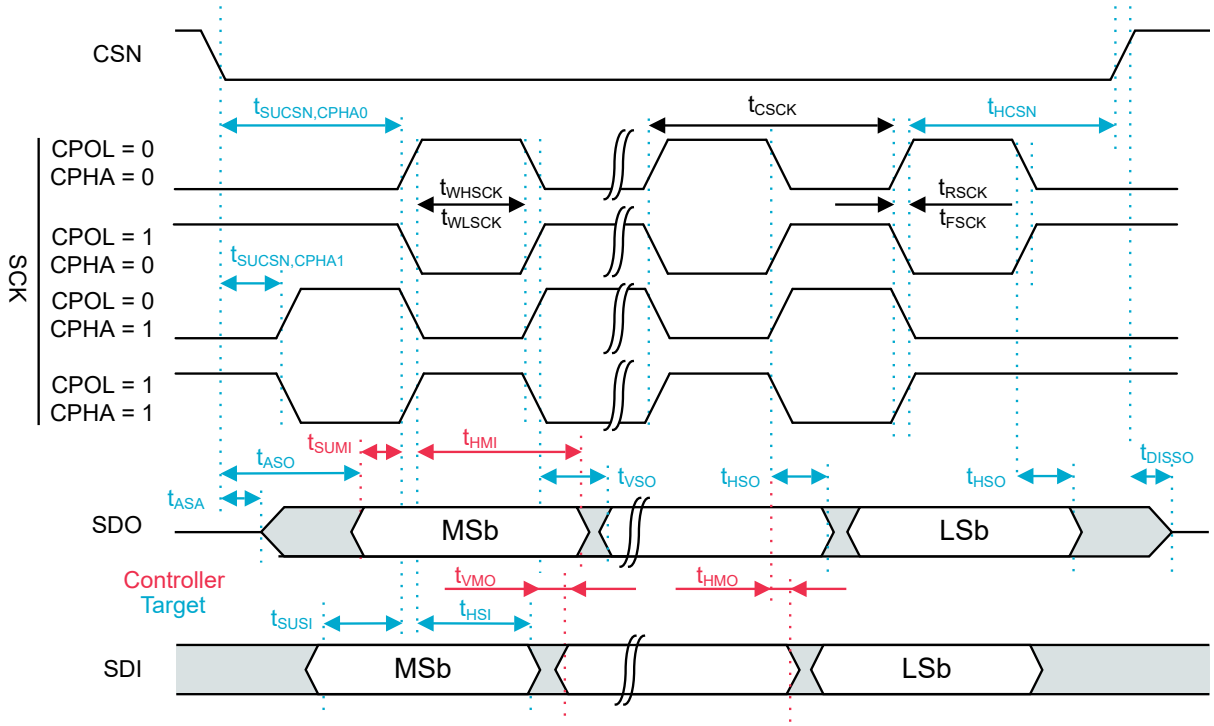


Figure 126: Common SPIM and SPIS timing diagram

11.9.3 Timing specifications for GPIO port P1 using Standard drive strength

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,VSO}$	CLK edge to SDO valid			35	ns
$t_{SPIS,HSO}$	SDO hold time after CLK edge	4			ns
$t_{SPIS,SUSI}$	SDI to CLK edge setup time	4			ns
$t_{SPIS,HSI}$	CLK edge to SDI hold time	2			ns

11.9.4 Timing specifications for GPIO port P1 using High drive strength

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,VSO}$	CLK edge to SDO valid			32	ns
$t_{SPIS,HSO}$	SDO hold time after CLK edge	4			ns
$t_{SPIS,SUSI}$	SDI to CLK edge setup time	4			ns
$t_{SPIS,HSI}$	CLK edge to SDI hold time	2			ns

11.10 SWDP Electrical specification

11.10.1 SW-DP

Symbol	Description	Min.	Typ.	Max.	Units
R_{pull}	Internal SWDIO and SWDCLK pull up/down resistance		14		kΩ
f_{SWDCLK}	SWDCLK frequency	0.125		20	MHz
t_{SUI}	SWDIO input setup before SWDCLK	18			ns
t_{HI}	SWDIO input hold after SWDCLK	1			ns
t_{VO}	SWDCLK to SWDIO output valid			35	ns
t_{HO}	SWDIO output hold after SWDCLK	4			ns

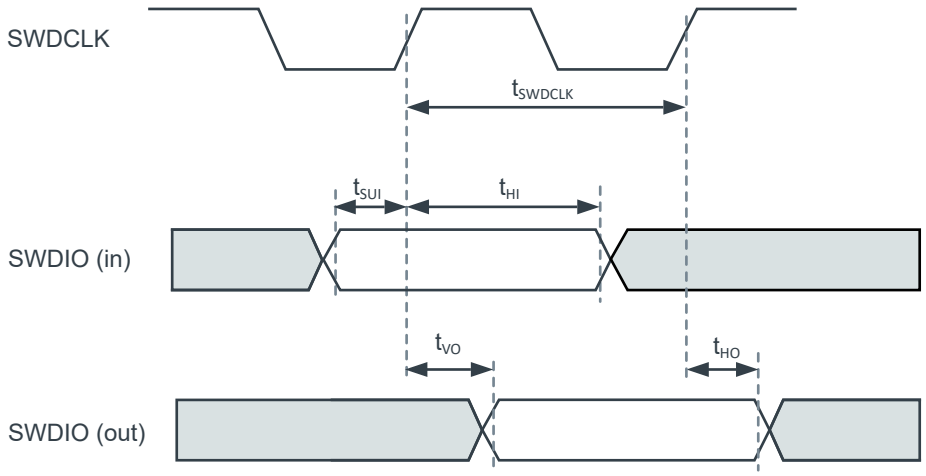


Figure 127: SWD timing diagram

11.11 TEMP Electrical specification

11.11.1 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
$T_{TEMP,RANGE,EXT}$	Temperature sensor extended temperature range	-40		85	°C
$T_{TEMP,ACC}$	Temperature sensor accuracy	-5		5	°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	-3		3	°C

11.12 TWIM Electrical specification

11.12.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL}$	Bit rates for TWIM[8]	100		1000	kbps
$t_{TWIM,START}$	Time from STARTRX/STARTTX task to transmission started		1.5		µs

11.12.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWIM,SU_DATI}	Input data setup time before positive edge on SCL – all modes	20			ns
t_{TWIM,HD_DATO}	Output data hold time after negative edge on SCL – 100, 250 and 400 kbps	500		625	ns
$t_{TWIM,HD\_STA,100kbps}$	TWIM master hold time for START and repeated START condition, 100 kbps	10000			ns
$t_{TWIM,HD\_STA,250kbps}$	TWIM master hold time for START and repeated START condition, 250 kbps	4000			ns
$t_{TWIM,HD\_STA,400kbps}$	TWIM master hold time for START and repeated START condition, 400 kbps	2400			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
$t_{R,100kbps}$	Rise time of both SDA and SCL signals, 100kbps			1000	ns
$t_{F,100kbps}$	Fall time of both SDA and SCL signals, 100kbps			300	ns
$t_{R,400kbps}$	Rise time of both SDA and SCL signals, 400kbps			300	ns
$t_{F,400kbps}$	Fall time of both SDA and SCL signals, 400kbps			300	ns
$t_{TWIM,BUF,100kbps}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5200			ns
$t_{TWIM,BUF,250kbps}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2200			ns
$t_{TWIM,BUF,400kbps}$	TWIM master bus free time between STOP and START conditions, 400 kbps	1500			ns

[8] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO — General purpose input/output on page 182 for more details.

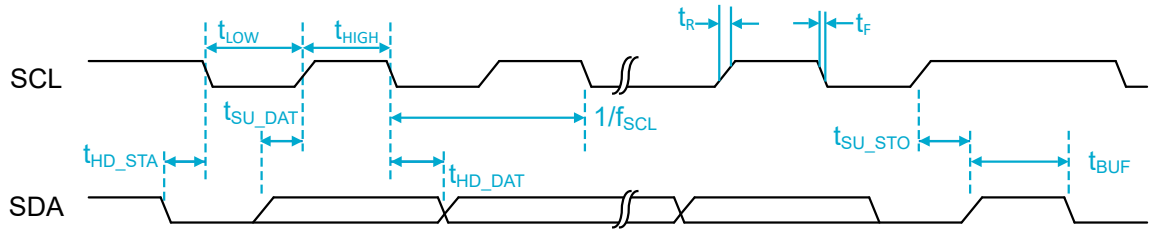


Figure 128: TWIM timing diagram, 1 byte transaction

11.13 TWIS Electrical specification

11.13.1 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIS,SCL}$	Bit rates for TWIS[9]	100		400	kbps
$t_{TWIS,START}$	Time from PREPARERX/PREPARETX task to ready to receive/transmit		1.5		µs
t_{TWIS,SU_DATI}	Input data setup time before positive edge on SCL – all modes	20			ns
t_{TWIS,HD_DATI}	Input data hold time after negative edge on SCL – all modes	0			ns
t_{TWIS,HD_DATO}	Output 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
$t_{TWIS,SU\_STO,400kbps}$	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

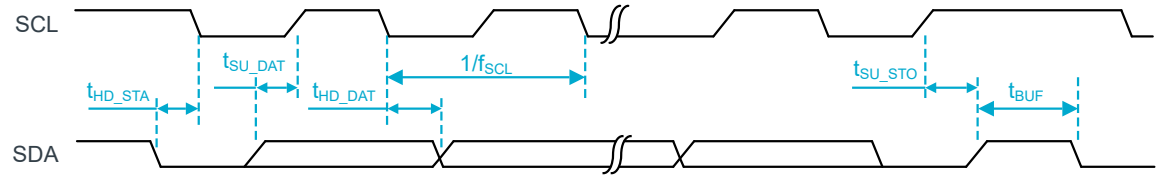


Figure 129: TWIS timing diagram, 1 byte transaction

11.14 WDT Electrical specification

11.14.1 Watchdog Timer Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{WDT}	Time out interval	458 µs		36 h	

[9] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

12 Recommended operating conditions

The operating conditions are the physical parameters that nRF54LS05A/nRF54LS05B can operate within.

Symbol	Parameter	Min.	Nom.	Max.	Units
VDD	VDD supply voltage	1.7		3.6	V
TA	Operating temperature	-40	25	85	°C

Table 60: Recommended operating conditions

13 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.

For accelerated lifetime testing (HTOL, etc.), supply voltage should not exceed the recommended operating conditions max value, see Recommended operating conditions on page 563.

	Parameter	Min.	Max.	Unit
VDD	VDD supply voltage	-0.3	3.9	V

Table 61: Supply voltage

	Parameter	Min.	Max.	Unit
$V_{I/O}$, VDD ≤3.6 V	IO voltage	-0.3	VDD + 0.3	V
$V_{I/O}$, VDD >3.6 V	IO voltage	-0.3	3.9	V

Table 62: I/O pin voltage

	Min.	Max.	Unit
RF input level		10	dBm

Table 63: Radio

	Note	Min.	Max.	Unit
Storage temperature		-40	+125	°C
Reflow soldering temperature	Reflow cycle time is 30 seconds with 3 maximum reflow cycles.		260	°C
Moisture Sensitivity Level (MSL)			TBD	
ESD Human Body Model (HBM)			TBD	kV
ESD Charged Device Model (CDM)			TBD	V

Table 64: Environmental QFN package types

	Min.	Max.	Unit
Endurance	10,000		Write/ rewrite cycles
Retention at 85°C	10		y

Table 65: RRAM memory



14 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 Datasheet prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

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.

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.

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 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.

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 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

© 2025 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.


QUALITY SYSTEM CERTIFICATION
DNV
ISO 9001

Contents

Feature overview . . . ii

1 Revision history . . . 3

2 About this document . . . 4

2.1 Document status . . . 4
2.2 Peripheral chapters . . . 4
2.2.1 Peripheral naming conventions . . . 4
2.3 Register overview table . . . 5
2.4 Register tables . . . 5
2.4.1 Fields and values . . . 5
2.4.2 Permissions . . . 6
2.5 Registers . . . 6
2.5.1 DUMMY . . . 6

3 Product overview . . . 8

3.1 Block diagram . . . 8
3.2 Memory and package overview . . . 9
3.3 Power domains . . . 10
3.4 Address format . . . 10
3.5 Memory . . . 11
3.5.1 RAM — Random access memory . . . 12
3.5.2 NVM — Non-volatile memory . . . 12
3.5.3 Memory map . . . 12
3.5.4 Instantiation . . . 13

4 Application core . . . 16

4.1 Arm Cortex-M33 CPU . . . 16
4.1.1 CPU . . . 16
4.1.2 CPUC — CPU control . . . 17
4.1.3 Arm Cortex-M33 Peripherals . . . 18
4.2 Core components . . . 19
4.2.1 AMBA interconnect (AMBIX) . . . 19
4.2.2 EasyDMA . . . 19
4.2.3 CACHE — Instruction/data cache . . . 21
4.2.4 FICR — Factory information configuration registers . . . 25
4.2.5 MEMCONF — Memory configuration . . . 30
4.2.6 RRAMC — Resistive random access memory controller . . . 32
4.2.7 SWI — Software interrupts . . . 45
4.2.8 UICR — User information configuration registers . . . 46

5 Power and clock management . . . 51

5.1 System ON mode . . . 51
5.1.1 Sub-power modes . . . 51
5.2 System OFF mode . . . 52
5.2.1 Emulated System OFF mode . . . 52
5.3 Power supply supervisors . . . 53
5.3.1 Power-fail comparator . . . 53
5.4 CLOCK — Clock control . . . 54

5.4.1 HFCLK controller . . . 55
5.4.2 LFCLK controller . . . 56
5.4.3 Registers . . . 57
5.5 OSCILLATORS — Oscillator control . . . 73
5.5.1 High-frequency (32 MHz) crystal oscillator (HFXO) . . . 73
5.5.2 Low-frequency (32.768 kHz) crystal oscillator (LFXO) . . . 75
5.5.3 CPU clock frequency selection . . . 76
5.5.4 Registers . . . 77
5.6 POWER — Power control . . . 79
5.6.1 Registers . . . 79
5.7 REGULATORS — Regulator control . . . 84
5.7.1 VREGMAIN — Main regulator . . . 85
5.7.2 Registers . . . 85
5.8 RESET — Reset control . . . 88
5.8.1 Power-on reset . . . 88
5.8.2 Pin reset . . . 89
5.8.3 Brownout reset . . . 89
5.8.4 Wakeup from System OFF mode reset . . . 89
5.8.5 Soft reset . . . 89
5.8.6 CTRL-AP resets . . . 89
5.8.7 Watchdog timer reset . . . 89
5.8.8 Retained registers . . . 89
5.8.9 Reset behavior . . . 90
5.8.10 Registers . . . 90

6 Event system . . . 93

6.1 DPPI latencies . . . 93
6.2 DPPI — Distributed programmable peripheral interconnect . . . 94
6.2.1 Channel publish and subscribe . . . 95
6.2.2 DPPI controller (DPPIC) . . . 97
6.2.3 Connection examples . . . 97
6.2.4 Registers . . . 98
6.3 PPIB — PPI Bridge . . . 102
6.3.1 PPIB connections . . . 102
6.3.2 Handshake and overflow . . . 103
6.3.3 Connection examples . . . 103
6.3.4 Registers . . . 104

7 Security . . . 107

7.1 Security components . . . 107
7.1.1 CRACEN TRNG — True Random Number Generator . . . 107
7.1.2 TAMPC — Tamper controller . . . 123

8 Peripherals . . . 138

8.1 Peripheral interface . . . 138
8.1.2 Peripheral ID . . . 139
8.1.3 Peripherals with shared ID . . . 139
8.1.4 Peripheral registers . . . 139
8.1.5 Bit set and clear . . . 140
8.1.6 Tasks . . . 140
8.1.7 Events . . . 140
8.1.8 Publish and subscribe . . . 140
8.1.9 Shortcuts . . . 140

8.1.10 Interrupts . . . 141
8.2 Instantiation . . . 141
8.3 AAR — Accelerated address resolver . . . 143
8.3.1 Shared resources . . . 143
8.3.2 Resolving a resolvable address . . . 143
8.3.3 EasyDMA . . . 146
8.3.4 Use case example for chaining RADIO packet reception with address resolution using AAR 147
8.3.5 Registers . . . 148
8.4 CCM — AES CCM mode encryption . . . 155
8.4.1 Shared resources . . . 156
8.4.2 Encryption and decryption . . . 156
8.4.3 Encrypting packets in radio transmit mode . . . 159
8.4.4 Decrypting packets received by the radio . . . 160
8.4.5 CCM data structure . . . 161
8.4.6 EasyDMA and ERROR event . . . 162
8.4.7 Registers . . . 162
8.5 ECB — AES electronic codebook mode encryption . . . 170
8.5.1 Shared resources . . . 172
8.5.2 EasyDMA . . . 172
8.5.3 Registers . . . 173
8.6 EGU — Event generator unit . . . 179
8.6.1 Registers . . . 179
8.7 GPIO — General purpose input/output . . . 182
8.7.1 Pin configuration . . . 182
8.7.2 Pin sense mechanism . . . 183
8.7.3 Port capabilities . . . 184
8.7.4 Peripheral and subsystem assignment . . . 185
8.7.5 Clock pins . . . 186
8.7.6 Weak pull-ups . . . 186
8.7.7 Reset behavior . . . 187
8.7.8 Registers . . . 187
8.8 GPIOTE — GPIO tasks and events . . . 192
8.8.1 Pin events and tasks . . . 193
8.8.2 Port event . . . 193
8.8.3 Tasks and events pin configuration . . . 194
8.8.4 Registers . . . 194
8.9 GRTC — Global real-time counter . . . 200
8.9.1 GRTC clock sources . . . 201
8.9.2 SYSCOUNTER . . . 201
8.9.3 Pulse Width Modulation (PWM) . . . 205
8.9.4 Clock output . . . 205
8.9.5 Task priority . . . 205
8.9.6 Registers . . . 206
8.10 PWM — Pulse width modulation . . . 227
8.10.1 Wave counter . . . 228
8.10.2 Decoder with EasyDMA . . . 231
8.10.3 Limitations . . . 239
8.10.4 Pin configuration . . . 239
8.10.5 Registers . . . 239
8.11 QDEC — Quadrature decoder . . . 259
8.11.1 Sampling and decoding . . . 260
8.11.2 LED output . . . 261
8.11.3 Debounce filters . . . 261
8.11.4 Accumulators . . . 261

8.11.5 Output/input pins . . . 262
8.11.6 Pin configuration . . . 262
8.11.7 Registers . . . 263
8.12 RADIO — 2.4 GHz radio . . . 277
8.12.1 Packet configuration . . . 278
8.12.2 Address configuration . . . 279
8.12.3 Data whitening . . . 280
8.12.4 CRC . . . 280
8.12.5 Radio states . . . 281
8.12.6 Transmit sequence . . . 283
8.12.7 Receive sequence . . . 284
8.12.8 Received signal strength indicator (RSSI) . . . 286
8.12.9 Interframe spacing (IFS) . . . 286
8.12.10 Device address match . . . 287
8.12.11 Bit counter . . . 288
8.12.12 EasyDMA . . . 288
8.12.13 Registers . . . 289
8.13 SAADC — Successive approximation analog-to-digital converter . . . 328
8.13.1 Overview . . . 328
8.13.2 Digital output . . . 329
8.13.3 Analog inputs and channels . . . 330
8.13.4 Operation modes . . . 330
8.13.5 EasyDMA . . . 332
8.13.6 Reference . . . 333
8.13.7 Acquisition time . . . 334
8.13.8 Limits event monitoring . . . 334
8.13.9 Performance factors . . . 335
8.13.10 Registers . . . 335
8.14 SPIM — Serial peripheral interface controller with EasyDMA . . . 353
8.14.1 SPIM transaction sequence . . . 354
8.14.2 D/CX functionality . . . 355
8.14.3 Chip select hardware control . . . 355
8.14.4 Pin configuration . . . 356
8.14.5 Shared resources . . . 357
8.14.6 EasyDMA . . . 357
8.14.7 Low power . . . 358
8.14.8 Registers . . . 358
8.15 SPIS — Serial peripheral interface target with EasyDMA . . . 380
8.15.1 SPI modes . . . 381
8.15.2 Shared resources . . . 381
8.15.3 EasyDMA . . . 382
8.15.4 SPIS operation . . . 382
8.15.5 Semaphore operation . . . 384
8.15.6 Pin configuration . . . 385
8.15.7 Registers . . . 385
8.16 TEMP — Temperature sensor . . . 404
8.16.1 Operation . . . 404
8.16.2 Registers . . . 405
8.17 TIMER — Timer/counter . . . 412
8.17.1 Capture . . . 414
8.17.2 Compare . . . 414
8.17.3 Task delays . . . 414
8.17.4 Task priority . . . 414
8.17.5 Registers . . . 415

8.18 TWIM — I²C compatible two-wire interface controller with EasyDMA . . . 421
8.18.1 TWIM operation . . . 422
8.18.2 Shared resources . . . 423
8.18.3 EasyDMA . . . 423
8.18.4 TWIM write sequence . . . 423
8.18.5 TWIM read sequence . . . 424
8.18.6 TWIM repeated start sequence . . . 425
8.18.7 Low power . . . 426
8.18.8 TWIM pin configuration . . . 426
8.18.9 Pull-up resistor . . . 427
8.18.10 Registers . . . 428
8.19 TWIS — I²C compatible two-wire interface target with EasyDMA . . . 452
8.19.1 State machine . . . 453
8.19.2 Shared resources . . . 454
8.19.3 EasyDMA . . . 454
8.19.4 Read command response . . . 455
8.19.5 Write command response . . . 456
8.19.6 TWI controller repeated start sequence . . . 457
8.19.7 Terminate an ongoing TWI transaction . . . 458
8.19.8 Low power . . . 458
8.19.9 Target mode pin configuration . . . 458
8.19.10 Registers . . . 459
8.20 UARTE — Universal asynchronous receiver/transmitter with EasyDMA . . . 481
8.20.1 Baudrate . . . 481
8.20.2 EasyDMA . . . 482
8.20.3 Transmission . . . 482
8.20.4 Reception . . . 483
8.20.5 Data frame size . . . 485
8.20.6 Frame timeout interrupt . . . 486
8.20.7 Error conditions . . . 487
8.20.8 Using the UARTE without flow control . . . 487
8.20.9 Parity and stop bit configuration . . . 487
8.20.10 Compare match filter . . . 488
8.20.11 Low power . . . 488
8.20.12 Pin configuration . . . 488
8.20.13 Registers . . . 489
8.21 WDT — Watchdog timer . . . 517
8.21.1 Reload criteria . . . 518
8.21.2 Temporarily pausing the watchdog . . . 518
8.21.3 Watchdog reset . . . 518
8.21.4 Stopping the watchdog . . . 518
8.21.5 Registers . . . 518

9 Debug . . . 526

9.1 Debug access port . . . 526
9.1.1 Debug access port registers . . . 527
9.2 Access port protection . . . 528
9.3 Debug Interface mode . . . 530
9.4 Real-time debug . . . 530
9.5 Multidrop serial wire debug . . . 531
9.6 CTRL-AP — Control access port . . . 531
9.6.1 Reset request . . . 532
9.6.2 Erase all . . . 532
9.6.3 Mailbox interface . . . 534

9.6.4 Device information . . . 535
9.6.5 Debugger registers . . . 535
9.6.6 Peripheral registers . . . 540
9.7 TAD — Trace and debug control . . . 544
9.7.1 Registers . . . 544

10 Hardware and layout . . . 546
10.1 Pin assignments . . . 546
10.1.1 Dedicated pins . . . 546
10.1.2 SWD pins . . . 546
10.1.3 Clock pins . . . 547
10.1.4 QFN48 (QFAA) package pin assignments . . . 547
10.2 Mechanical specifications . . . 551
10.2.1 QFN48 (QFAA) package . . . 551
10.3 Reference circuitry . . . 552
10.4 Reflow conditions . . . 552

11 Electrical specifications . . . 553
11.1 Current consumption . . . 553
11.1.1 CURRENT Electrical specification . . . 553
11.2 CLOCK Electrical specification . . . 554
11.2.1 High frequency clock source (HFCLK) . . . 554
11.2.2 32.768 kHz clock source (LFCLK) . . . 554
11.3 CPU Electrical specification . . . 554
11.3.1 CPU performance . . . 554
11.4 GPIO Electrical specification . . . 555
11.4.1 GPIO Electrical Specification . . . 555
11.5 OSCILLATORS Electrical specification . . . 555
11.5.1 32 MHz crystal oscillator (HFXO) . . . 555
11.5.2 32.768 kHz crystal oscillator (LFXO) . . . 555
11.6 REGULATORS Electrical specification . . . 555
11.6.1 Recommended operating conditions . . . 555
11.6.2 Power-fail comparator . . . 556
11.7 RESET Electrical specification . . . 556
11.7.1 Startup times . . . 556
11.8 SPIM Electrical specification . . . 556
11.8.1 Timing specifications . . . 556
11.8.2 Timing specifications for GPIO port P1 using Standard drive strength . . . 557
11.8.3 Timing specifications for GPIO port P1 using High drive strength . . . 557
11.9 SPIS Electrical specification . . . 558
11.9.1 SPIS slave interface electrical specifications . . . 558
11.9.2 Serial Peripheral Interface Slave (SPIS) timing specifications . . . 558
11.9.3 Timing specifications for GPIO port P1 using Standard drive strength . . . 559
11.9.4 Timing specifications for GPIO port P1 using High drive strength . . . 560
11.10 SWDP Electrical specification . . . 560
11.10.1 SW-DP . . . 560
11.11 TEMP Electrical specification . . . 561
11.11.1 Temperature Sensor Electrical Specification . . . 561
11.12 TWIM Electrical specification . . . 561
11.12.1 TWIM interface electrical specifications . . . 561
11.12.2 Two Wire Interface Master (TWIM) timing specifications . . . 561
11.13 TWIS Electrical specification . . . 562
11.13.1 TWIS slave timing specifications . . . 562

11.14 WDT Electrical specification . . . 562
11.14.1 Watchdog Timer Electrical Specification . . . 562

12 Recommended operating conditions . . . 563

13 Absolute maximum ratings . . . 564

14 Legal notices . . . 566