In this mode, the RESULTDONE event signals has the same meaning as DONE when no oversampling takes place. Note that both events may occur before the actual values have been transferred into RAM by EasyDMA.

Example of RAM placement (even RESULT.MAXCNT), channels 1, 2 and 5 enabled on page 199 provides an example of results placement in Data RAM, with an even RESULT.MAXCNT. In this example, channels 1, 2 and 5 are enabled, all others are disabled.

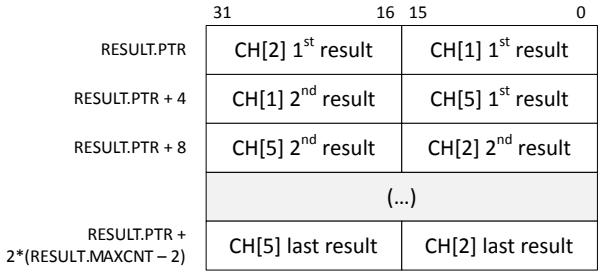

Figure 63: Example of RAM placement (even RESULT.MAXCNT), channels 1, 2 and 5 enabled

Example of RAM placement (odd RESULT.MAXCNT), channels 1, 2 and 5 enabled on page 199 provides an example of results placement in Data RAM, with an odd RESULT.MAXCNT. In this example, channels 1, 2 and 5 are enabled, all others are disabled. The last 32-bit word is populated only with one 16-bit result.

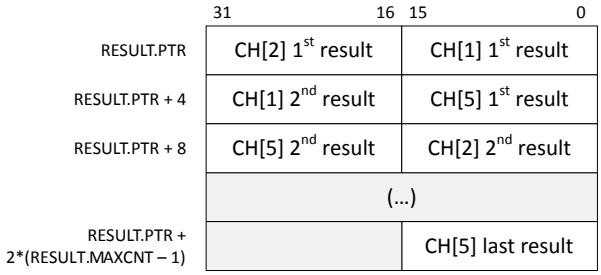

Figure 64: Example of RAM placement (odd RESULT.MAXCNT), channels 1, 2 and 5 enabled

6.14.6 EasyDMA

After configuring RESULT.PTR and RESULT.MAXCNT, the ADC resources are started by triggering the START task. The ADC is using EasyDMA to store results in a Result buffer in RAM.

The Result buffer is located at the address specified in the RESULT.PTR register. The RESULT.PTR register is double-buffered and it can be updated and prepared for the next START task immediately after the STARTED event is generated. The size of the Result buffer is specified in the RESULT.MAXCNT register and the ADC will generate an END event when it has filled up the Result buffer, see ADC on page 200. 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.

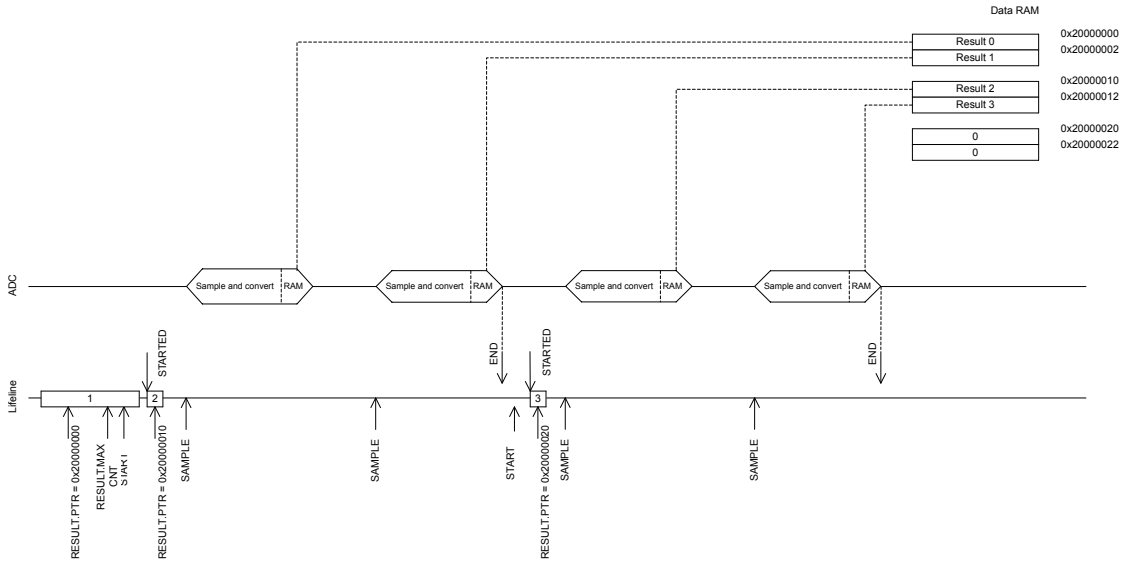

Figure 65: ADC

If the RESULT.PTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

The EasyDMA will have finished accessing the RAM when the END or STOPPED event has been generated.

The RESULT.AMOUNT register can be read following an END event or a STOPPED event to see how many results have been transferred to the Result buffer in RAM since the START task was triggered.

In scan mode, SAMPLE tasks can be triggered once the START task is triggered. The END event is generated when the number of samples transferred to memory reaches the value specified by RESULT.MAXCNT. After an END event, the START task needs to be triggered again before new samples can be taken. Also make sure that the size of the Result buffer is large enough to have space for minimum one result from each of the enabled channels, by specifying RESULT.MAXCNT >= number of channels enabled. For more information about the scan mode, see Scan mode on page 198.

6.14.7 Resistor ladder

The ADC has an internal resistor string for positive and negative input.

See Resistor ladder for positive input (negative input is equivalent, using RESN instead of RESP) on page 201. The resistors are controlled in the CH[n].CONFIG.RESP and CH[n].CONFIG.RESN registers.

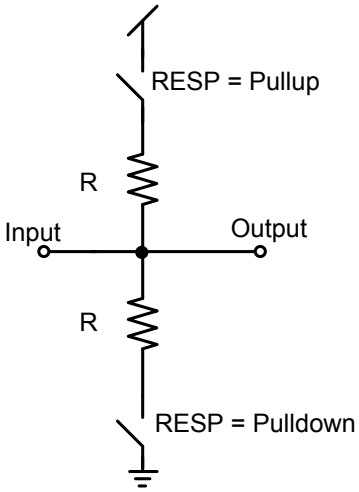

Figure 66: Resistor ladder for positive input (negative input is equivalent, using RESN instead of RESP)

6.14.8 Reference

The ADC can use two different references, controlled in the REFSEL field of the CH[n].CONFIG register.

These are:

- Internal reference
- VDD as reference

The internal reference results in an input range of ±0.6 V on the ADC core. VDD as reference results in an input range of ±VDD/4 on the ADC core. The gain block can be used to change the effective input range of the ADC.

```
Input range = (+- 0.6 V or +-VDD/4)/Gain
```

For example, choosing VDD as reference, single ended input (grounded negative input), and a gain of 1/4 the input range will be:

```
Input range = (VDD/4)/(1/4) = VDD
```

With internal reference, single ended input (grounded negative input), and a gain of 1/6 the input range will be:

```
Input range = (0.6 V)/(1/6) = 3.6 V
```

The AIN0-AIN7 inputs cannot exceed VDD, or be lower than VSS.

6.14.9 Acquisition time

To sample the input voltage, the ADC connects a capacitor to the input.

For illustration, see Simplified ADC sample network on page 202. 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 202.

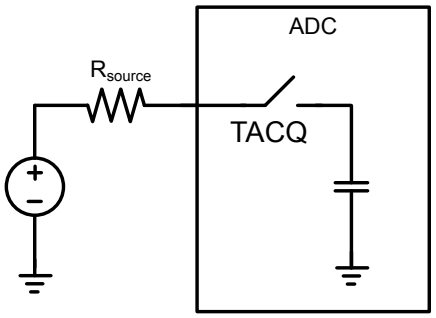

Figure 67: Simplified ADC sample network

TACQ [µs]	Maximum source resistance [kOhm]
3	10
5	40
10	100
15	200
20	400
40	800

Table 64: Acquisition time

6.14.10 Limits event monitoring

A channel can be event monitored by configuring limit register CH[n].LIMIT.

If the conversion result is higher than the defined high limit, or lower than the defined low limit, the appropriate event will get fired.

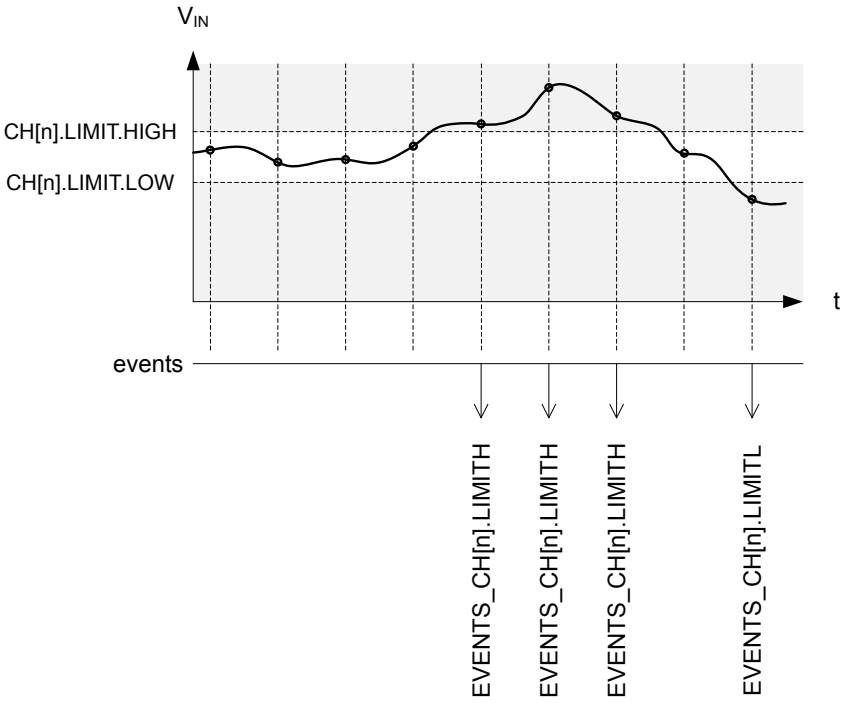

Figure 68: Example of limits monitoring on channel 'n'

Note that when setting the limits, CH[n].LIMIT.HIGH shall always be higher than or equal to CH[n].LIMIT.LOW . In other words, an event can be fired only when the input signal has been sampled outside of the defined limits. It is not possible to fire an event when the input signal is inside a defined range by swapping high and low limits.

The comparison to limits always takes place, there is no need to enable it. If comparison is not required on a channel, the software shall simply ignore the related events. In that situation, the value of the limits registers is irrelevant, so it does not matter if CH[n].LIMIT.LOW is lower than CH[n].LIMIT.HIGH or not.

6.14.11 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5000E000 0x4000E000	SAADC	SAADC : S SAADC : NS	US	SA	Analog to digital converter	

Table 65: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start the ADC and prepare the result buffer in RAM
TASKS_SAMPLE	0x004		Take one ADC sample, if scan is enabled all channels are sampled
TASKS_STOP	0x008		Stop the ADC and terminate any 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_CALIBRATEOFF	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[0].LIMITH	0x118		Last results is equal or above CH[0].LIMIT.HIGH
EVENTS_CH[0].LIMITL	0x11C		Last results is equal or below CH[0].LIMIT.LOW
EVENTS_CH[1].LIMITH	0x120		Last results is equal or above CH[1].LIMIT.HIGH
EVENTS_CH[1].LIMITL	0x124		Last results is equal or below CH[1].LIMIT.LOW
EVENTS_CH[2].LIMITH	0x128		Last results is equal or above CH[2].LIMIT.HIGH
EVENTS_CH[2].LIMITL	0x12C		Last results is equal or below CH[2].LIMIT.LOW
EVENTS_CH[3].LIMITH	0x130		Last results is equal or above CH[3].LIMIT.HIGH
EVENTS_CH[3].LIMITL	0x134		Last results is equal or below CH[3].LIMIT.LOW
EVENTS_CH[4].LIMITH	0x138		Last results is equal or above CH[4].LIMIT.HIGH
EVENTS_CH[4].LIMITL	0x13C		Last results is equal or below CH[4].LIMIT.LOW
EVENTS_CH[5].LIMITH	0x140		Last results is equal or above CH[5].LIMIT.HIGH
EVENTS_CH[5].LIMITL	0x144		Last results is equal or below CH[5].LIMIT.LOW
EVENTS_CH[6].LIMITH	0x148		Last results is equal or above CH[6].LIMIT.HIGH
EVENTS_CH[6].LIMITL	0x14C		Last results is equal or below CH[6].LIMIT.LOW
EVENTS_CH[7].LIMITH	0x150		Last results is equal or above CH[7].LIMIT.HIGH
EVENTS_CH[7].LIMITL	0x154		Last results is equal or below CH[7].LIMIT.LOW
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

Register	Offset	Security	Description
PUBLISH_STOPPED	0x194		Publish configuration for event STOPPED
PUBLISH_CH[0].LIMITH	0x198		Publish configuration for event CH[0].LIMITH
PUBLISH_CH[0].LIMITL	0x19C		Publish configuration for event CH[0].LIMITL
PUBLISH_CH[1].LIMITH	0x1A0		Publish configuration for event CH[1].LIMITH
PUBLISH_CH[1].LIMITL	0x1A4		Publish configuration for event CH[1].LIMITL
PUBLISH_CH[2].LIMITH	0x1A8		Publish configuration for event CH[2].LIMITH
PUBLISH_CH[2].LIMITL	0x1AC		Publish configuration for event CH[2].LIMITL
PUBLISH_CH[3].LIMITH	0x1B0		Publish configuration for event CH[3].LIMITH
PUBLISH_CH[3].LIMITL	0x1B4		Publish configuration for event CH[3].LIMITL
PUBLISH_CH[4].LIMITH	0x1B8		Publish configuration for event CH[4].LIMITH
PUBLISH_CH[4].LIMITL	0x1BC		Publish configuration for event CH[4].LIMITL
PUBLISH_CH[5].LIMITH	0x1C0		Publish configuration for event CH[5].LIMITH
PUBLISH_CH[5].LIMITL	0x1C4		Publish configuration for event CH[5].LIMITL
PUBLISH_CH[6].LIMITH	0x1C8		Publish configuration for event CH[6].LIMITH
PUBLISH_CH[6].LIMITL	0x1CC		Publish configuration for event CH[6].LIMITL
PUBLISH_CH[7].LIMITH	0x1D0		Publish configuration for event CH[7].LIMITH
PUBLISH_CH[7].LIMITL	0x1D4		Publish configuration for event CH[7].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[0].PSELP	0x510		Input positive pin selection for CH[0]
CH[0].PSELN	0x514		Input negative pin selection for CH[0]
CH[0].CONFIG	0x518		Input configuration for CH[0]
CH[0].LIMIT	0x51C		High/low limits for event monitoring a channel
CH[1].PSELP	0x520		Input positive pin selection for CH[1]
CH[1].PSELN	0x524		Input negative pin selection for CH[1]
CH[1].CONFIG	0x528		Input configuration for CH[1]
CH[1].LIMIT	0x52C		High/low limits for event monitoring a channel
CH[2].PSELP	0x530		Input positive pin selection for CH[2]
CH[2].PSELN	0x534		Input negative pin selection for CH[2]
CH[2].CONFIG	0x538		Input configuration for CH[2]
CH[2].LIMIT	0x53C		High/low limits for event monitoring a channel
CH[3].PSELP	0x540		Input positive pin selection for CH[3]
CH[3].PSELN	0x544		Input negative pin selection for CH[3]
CH[3].CONFIG	0x548		Input configuration for CH[3]
CH[3].LIMIT	0x54C		High/low limits for event monitoring a channel
CH[4].PSELP	0x550		Input positive pin selection for CH[4]
CH[4].PSELN	0x554		Input negative pin selection for CH[4]
CH[4].CONFIG	0x558		Input configuration for CH[4]
CH[4].LIMIT	0x55C		High/low limits for event monitoring a channel
CH[5].PSELP	0x560		Input positive pin selection for CH[5]
CH[5].PSELN	0x564		Input negative pin selection for CH[5]
CH[5].CONFIG	0x568		Input configuration for CH[5]
CH[5].LIMIT	0x56C		High/low limits for event monitoring a channel
CH[6].PSELP	0x570		Input positive pin selection for CH[6]
CH[6].PSELN	0x574		Input negative pin selection for CH[6]
CH[6].CONFIG	0x578		Input configuration for CH[6]
CH[6].LIMIT	0x57C		High/low limits for event monitoring a channel
CH[7].PSELP	0x580		Input positive pin selection for CH[7]
CH[7].PSELN	0x584		Input negative pin selection for CH[7]
CH[7].CONFIG	0x588		Input configuration for CH[7]

Register	Offset	Security	Description
CH[7].LIMIT	0x58C		High/low limits for event monitoring a channel
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 words to transfer
RESULT.AMOUNT	0x634		Number of buffer words transferred since last START

Table 66: Register overview

6.14.11.1 TASKS_START

Address offset: 0x000

Start the ADC and prepare the result buffer in RAM

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_START			Start the ADC and prepare the result buffer in RAM
			Trigger	1	Trigger task

6.14.11.2 TASKS_SAMPLE

Address offset: 0x004

Take one ADC sample, if scan is enabled all channels are sampled

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SAMPLE			Take one ADC sample, if scan is enabled all channels are sampled
			Trigger	1	Trigger task

6.14.11.3 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop the ADC and terminate any on-going conversion
			Trigger	1	Trigger task

6.14.11.4 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_CALIBRATEOFFSET			Starts offset auto-calibration
			Trigger	1	Trigger task

6.14.11.5 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task START

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

6.14.11.6 SUBSCRIBE_SAMPLE

Address offset: 0x084

Subscribe configuration for task SAMPLE

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

6.14.11.7 SUBSCRIBE_STOP

Address offset: 0x088

Subscribe configuration for task STOP

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

6.14.11.8 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task CALIBRATEOFFSET will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.14.11.9 EVENTS_STARTED

Address offset: 0x100

The ADC has started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			The ADC has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.14.11.10 EVENTS_END

Address offset: 0x104

The ADC has filled up the Result buffer

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

6.14.11.11 EVENTS_DONE

Address offset: 0x108

A conversion task has been completed. Depending on the mode, multiple conversions might be needed for a result to be transferred to RAM.

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

6.14.11.12 EVENTS_RESULTDONE

Address offset: 0x10C

A result is ready to get transferred to RAM.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RESULTDONE			A result is ready to get transferred to RAM.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.14.11.13 EVENTS_CALIBRATEDONE

Address offset: 0x110

Calibration is complete

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_CALIBRATEDONE			Calibration is complete
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.14.11.14 EVENTS_STOPPED

Address offset: 0x114

The ADC has stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			The ADC has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.14.11.15 EVENTS_CH[n].LIMITH (n=0..7)

Address offset: 0x118 + (n × 0x8)

Last results is equal or above CH[n].LIMIT.HIGH

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

ID	Acce	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

6.14.11.16 EVENTS_CH[n].LIMITL (n=0..7)

Address offset: 0x11C + (n × 0x8)

Last results is equal or below CH[n].LIMIT.LOW

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

ID	Acce	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

6.14.11.17 PUBLISH_STARTED

Address offset: 0x180

Publish configuration for event STARTED

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event STARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.14.11.18 PUBLISH_END

Address offset: 0x184

Publish configuration for event END

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event END will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.14.11.19 PUBLISH_DONE

Address offset: 0x188

Publish configuration for event DONE

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

6.14.11.20 PUBLISH_RESULTDONE

Address offset: 0x18C

Publish configuration for event RESULTDONE

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

6.14.11.21 PUBLISH_CALIBRATEDONE

Address offset: 0x190

Publish configuration for event CALIBRATEDONE

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

6.14.11.22 PUBLISH_STOPPED

Address offset: 0x194

Publish configuration for event STOPPED

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.14.11.23 PUBLISH_CH[n].LIMITH (n=0..7)

Address offset: 0x198 + (n × 0x8)

Publish configuration for event CH[n].LIMITH

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event CH[n].LIMITH will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.14.11.24 PUBLISH_CH[n].LIMITL (n=0..7)

Address offset: 0x19C + (n × 0x8)

Publish configuration for event CH[n].LIMITL

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event CH[n].LIMITL will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.14.11.25 INTEN

Address offset: 0x300

Enable or disable interrupt

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

ID	Acce	Field	Value ID	Value	Description
A	RW	STARTED			Enable or disable interrupt for event STARTED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	END			Enable or disable interrupt for event END
			Disabled	0	Disable
			Enabled	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
C	RW	DONE			Enable or disable interrupt for event DONE
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	RESULTDONE			Enable or disable interrupt for event RESULTDONE
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	CALIBRATEDONE			Enable or disable interrupt for event CALIBRATEDONE
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	CH0LIMITH			Enable or disable interrupt for event CH0LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	CH0LIMITL			Enable or disable interrupt for event CH0LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	CH1LIMITH			Enable or disable interrupt for event CH1LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	CH1LIMITL			Enable or disable interrupt for event CH1LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
K	RW	CH2LIMITH			Enable or disable interrupt for event CH2LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	CH2LIMITL			Enable or disable interrupt for event CH2LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
M	RW	CH3LIMITH			Enable or disable interrupt for event CH3LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
N	RW	CH3LIMITL			Enable or disable interrupt for event CH3LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
O	RW	CH4LIMITH			Enable or disable interrupt for event CH4LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
P	RW	CH4LIMITL			Enable or disable interrupt for event CH4LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
Q	RW	CH5LIMITH			Enable or disable interrupt for event CH5LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
R	RW	CH5LIMITL			Enable or disable interrupt for event CH5LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
S	RW	CH6LIMITH			Enable or disable interrupt for event CH6LIMITH
			Disabled	0	Disable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Enabled	1	Enable
T	RW	CH6LIMITL			Enable or disable interrupt for event CH6LIMITL
			Disabled	0	Disable
			Enabled	1	Enable
U	RW	CH7LIMITH			Enable or disable interrupt for event CH7LIMITH
			Disabled	0	Disable
			Enabled	1	Enable
V	RW	CH7LIMITL			Enable or disable interrupt for event CH7LIMITL
			Disabled	0	Disable
			Enabled	1	Enable

6.14.11.26 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to enable interrupt for event STARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to enable interrupt for event DONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to enable interrupt for event RESULTDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to enable interrupt for event CALIBRATEDONE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to enable interrupt for event CH0LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to enable interrupt for event CH0LIMITL

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to enable interrupt for event CH1LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to enable interrupt for event CH1LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to enable interrupt for event CH2LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to enable interrupt for event CH2LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to enable interrupt for event CH3LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to enable interrupt for event CH3LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	CH4LIMITH			Write '1' to enable interrupt for event CH4LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to enable interrupt for event CH4LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to enable interrupt for event CH5LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to enable interrupt for event CH5LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to enable interrupt for event CH6LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to enable interrupt for event CH6LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Enabled	1	Read: Enabled
U	RW	CH7LIMITH			Write '1' to enable interrupt for event CH7LIMITH
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to enable interrupt for event CH7LIMITL
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.14.11.27 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STARTED			Write '1' to disable interrupt for event STARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	DONE			Write '1' to disable interrupt for event DONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	RESULTDONE			Write '1' to disable interrupt for event RESULTDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	CALIBRATEDONE			Write '1' to disable interrupt for event CALIBRATEDONE
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	CH0LIMITH			Write '1' to disable interrupt for event CH0LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	CH0LIMITL			Write '1' to disable interrupt for event CH0LIMITL
			Clear	1	Disable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	CH1LIMITH			Write '1' to disable interrupt for event CH1LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	CH1LIMITL			Write '1' to disable interrupt for event CH1LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
K	RW	CH2LIMITH			Write '1' to disable interrupt for event CH2LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	CH2LIMITL			Write '1' to disable interrupt for event CH2LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
M	RW	CH3LIMITH			Write '1' to disable interrupt for event CH3LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
N	RW	CH3LIMITL			Write '1' to disable interrupt for event CH3LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
O	RW	CH4LIMITH			Write '1' to disable interrupt for event CH4LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
P	RW	CH4LIMITL			Write '1' to disable interrupt for event CH4LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
Q	RW	CH5LIMITH			Write '1' to disable interrupt for event CH5LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
R	RW	CH5LIMITL			Write '1' to disable interrupt for event CH5LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
S	RW	CH6LIMITH			Write '1' to disable interrupt for event CH6LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
T	RW	CH6LIMITL			Write '1' to disable interrupt for event CH6LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				V U T S R Q P O N M L K J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
U	RW	CH7LIMITH			Write '1' to disable interrupt for event CH7LIMITH
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
V	RW	CH7LIMITL			Write '1' to disable interrupt for event CH7LIMITL
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.14.11.28 STATUS

Address offset: 0x400

Status

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	STATUS			Status
			Ready	0	ADC is ready. No on-going conversion.
			Busy	1	ADC is busy. Conversion in progress.

6.14.11.29 ENABLE

Address offset: 0x500

Enable or disable ADC

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable ADC
			Disabled	0	Disable ADC
			Enabled	1	Enable ADC
					When enabled, the ADC will acquire access to the analog input pins specified in the CH[n].PSELP and CH[n].PSELN registers.

6.14.11.30 CH[n].PSELP (n=0..7)

Address offset: 0x510 + (n × 0x10)

Input positive pin selection for CH[n]

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PSELP			Analog positive input channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD

6.14.11.31 CH[n].PSELN (n=0..7)

Address offset: 0x514 + (n × 0x10)

Input negative pin selection for CH[n]

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PSELN			Analog negative input, enables differential channel
			NC	0	Not connected
			AnalogInput0	1	AIN0
			AnalogInput1	2	AIN1
			AnalogInput2	3	AIN2
			AnalogInput3	4	AIN3
			AnalogInput4	5	AIN4
			AnalogInput5	6	AIN5
			AnalogInput6	7	AIN6
			AnalogInput7	8	AIN7
			VDD	9	VDD

6.14.11.32 CH[n].CONFIG (n=0..7)

Address offset: 0x518 + (n × 0x10)

Input configuration for CH[n]

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			G F E E E D C C C B B A A
Reset 0x00020000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	RESP			Positive channel resistor control
			Bypass	0	Bypass resistor ladder
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
B	RW	RESN			Negative channel resistor control
			Bypass	0	Bypass resistor ladder

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				G F E E E D C C C B B A A
Reset 0x00020000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Pulldown	1	Pull-down to GND
			Pullup	2	Pull-up to VDD
			VDD1_2	3	Set input at VDD/2
C	RW	GAIN			Gain control
			Gain1_6	0	1/6
			Gain1_5	1	1/5
			Gain1_4	2	1/4
			Gain1_3	3	1/3
			Gain1_2	4	1/2
			Gain1	5	1
			Gain2	6	2
			Gain4	7	4
D	RW	REFSEL			Reference control
			Internal	0	Internal reference (0.6 V)
			VDD1_4	1	VDD/4 as reference
E	RW	TACQ			Acquisition time, the time the ADC uses to sample the input voltage
			3us	0	3 us
			5us	1	5 us
			10us	2	10 us
			15us	3	15 us
			20us	4	20 us
			40us	5	40 us
F	RW	MODE			Enable differential mode
			SE	0	Single ended, PSELN will be ignored, negative input to ADC shorted to GND
			Diff	1	Differential
G	RW	BURST			Enable burst mode
			Disabled	0	Burst mode is disabled (normal operation)
			Enabled	1	Burst mode is enabled. SAADC takes 2^OVERSAMPLE number of samples as fast as it can, and sends the average to Data RAM.

6.14.11.33 CH[n].LIMIT (n=0..7)

Address offset: 0x51C + (n × 0x10)

High/low limits for event monitoring a channel

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B B B B B B B B B B B B B B B B A A A A A A A A A A A A A A
Reset 0x7FFF8000				**0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LOW		[-32768 to +32767]	Low level limit
B	RW	HIGH		[-32768 to +32767]	High level limit

6.14.11.34 RESOLUTION

Address offset: 0x5F0

Resolution configuration

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

ID	Acce	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

6.14.11.35 OVERSAMPLE

Address offset: 0x5F4

Oversampling configuration. OVERSAMPLE should not be combined with SCAN. The RESOLUTION is applied before averaging, thus for high OVERSAMPLE a higher RESOLUTION should be used.

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

ID	Acce	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

6.14.11.36 SAMPLERATE

Address offset: 0x5F8

Controls normal or continuous sample rate

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CC		[80..2047]	Capture and compare value. Sample rate is 16 MHz/CC
B	RW	MODE			Select mode for sample rate control
			Task	0	Rate is controlled from SAMPLE task
			Timers	1	Rate is controlled from local timer (use CC to control the rate)

6.14.11.37 RESULT.PTR

Address offset: 0x62C

Data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.14.11.38 RESULT.MAXCNT

Address offset: 0x630

Maximum number of buffer words to transfer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT			Maximum number of buffer words to transfer

6.14.11.39 RESULT.AMOUNT

Address offset: 0x634

Number of buffer words transferred since last START

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

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT			Number of buffer words transferred since last START. This register can be read after an END or STOPPED event.

6.14.12 Electrical specification

6.14.12.1 SAADC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
DNL_{10}	Differential non-linearity, 10-bit resolution	-0.95	<1		LSB10b
INL_{10}	Integral non-linearity, 10-bit resolution		1		LSB1(
V_{OS}	Differential offset error (calibrated), 10-bit resolution [a]	-15		15	LSB10b
DNL_{12}	Differential non-linearity, 12-bit resolution	..	..	..	LSB1:
INL_{12}	Integral non-linearity, 12-bit resolution	..	..	..	LSB12b
C_{EG}	Gain error temperature coefficient	-0.05		0.05	%/°C
f_{SAMPLE}	Maximum sampling rate			200	kHz
$t_{ACQ,10k}$	Acquisition time (configurable), source Resistance <= 10kOhm		3		µs
$t_{ACQ,40k}$	Acquisition time (configurable), source Resistance <= 40kOhm		5		µs

[a] Digital output code at zero volt differential input.

Symbol	Description	Min.	Typ.	Max.	Units
$t_{ACQ,100k}$	Acquisition time (configurable), source Resistance <= 100kOhm		10		µs
$t_{ACQ,200k}$	Acquisition time (configurable), source Resistance <= 200kOhm		15		µs
$t_{ACQ,400k}$	Acquisition time (configurable), source Resistance <= 400kOhm		20		µs
$t_{ACQ,800k}$	Acquisition time (configurable), source Resistance <= 800kOhm		40		µs
t_{CONV}	Conversion time		<2		µs
$E_{G1/6}$	Error[b] for Gain = 1/6	-3		3	%
$E_{G1/4}$	Error[b] for Gain = 1/4	-3		3	%
$E_{G1/2}$	Error[b] for Gain = 1/2	-3		4	%
E_{G1}	Error[b] for Gain = 1	-3		4	%
C_{SAMPLE}	Sample and hold capacitance at maximum gain[9]		2.5		pF
R_{INPUT}	Input resistance		>1		MΩ
E_{NOB}	Effective number of bits, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		9		Bit
S_{NDR}	Peak signal to noise and distortion ratio, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		56		dB
S_{FDR}	Spurious free dynamic range, differential mode, 12-bit resolution, 1/1 gain, 3 µs acquisition time, crystal HFCLK, 200 ksps		70		dBc
R_{LADDER}	Ladder resistance		160		kΩ

6.14.13 Performance factors

Clock jitter, affecting sample timing accuracy, and circuit noise can affect ADC performance.

Jitter can be between START tasks or from START task to acquisition. START timer accuracy and startup times of regulators and references will contribute to variability. Sources of circuit noise may include CPU activity and the DC/DC regulator. Best ADC performance is achieved using START timing based on the TIMER module, HFXO clock source, and Constant Latency mode.

6.15 SPIM — Serial peripheral interface master with EasyDMA

The SPI master can communicate with multiple slaves using individual chip select signals for each of the slave devices attached to a bus.

Listed here are the main features for the SPIM

- SPI mode 0-3
- EasyDMA direct transfer to/from RAM for both SPI Slave and SPI Master
- Individual selection of IO pin for each SPI signal

[b] Does not include temperature drift

[9] Maximum gain corresponds to highest capacitance.

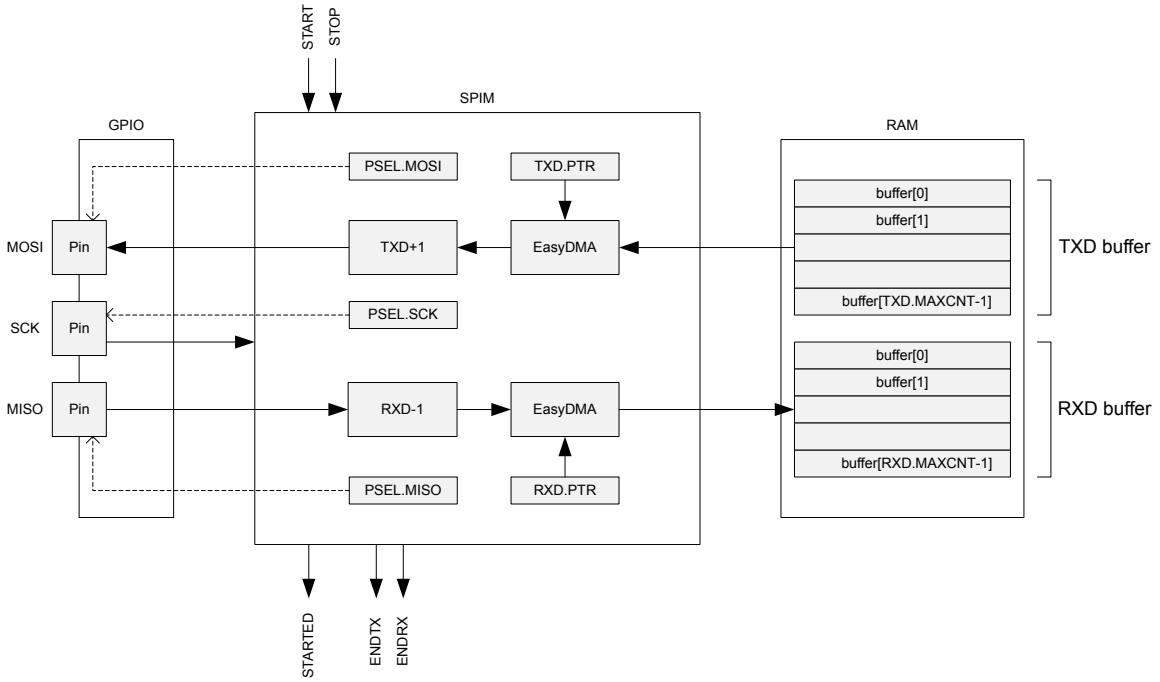

Figure 69: SPIM — SPI master with EasyDMA

The SPIM does not implement support for chip select directly. Therefore, the CPU must use available GPIOs to select the correct slave and control this independently of the SPI master. The SPIM supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Active High)	0 (Leading)
SPI_MODE1	0 (Active High)	1 (Trailing)
SPI_MODE2	1 (Active Low)	0 (Leading)
SPI_MODE3	1 (Active Low)	1 (Trailing)

Table 67: SPI modes

6.15.1 SPI master transaction sequence

An SPI master transaction consists of a sequence started by the START task followed by a number of events, and finally the STOP task.

An SPI master transaction is started by triggering the START task. The ENDTX event will be generated when the transmitter has transmitted all bytes in the TXD buffer as specified in the TXD.MAXCNT register. The ENDRX event will be generated when the receiver has filled the RXD buffer, i.e. received the last possible byte as specified in the RXD.MAXCNT register.

Following a START task, the SPI master will generate an END event when both ENDRX and ENDTX have been generated.

The SPI master is stopped by triggering the STOP task. A STOPPED event is generated when the SPI master has stopped.

If the ENDRX event has not already been generated when the SPI master has come to a stop, the SPI master will generate the ENDRX event explicitly even though the RX buffer is not full.

If the ENDTX event has not already been generated when the SPI master has come to a stop, the SPI master will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

The SPI master is a synchronous interface, and for every byte that is sent, a different byte will be received at the same time; this is illustrated in SPI master transaction on page 224.

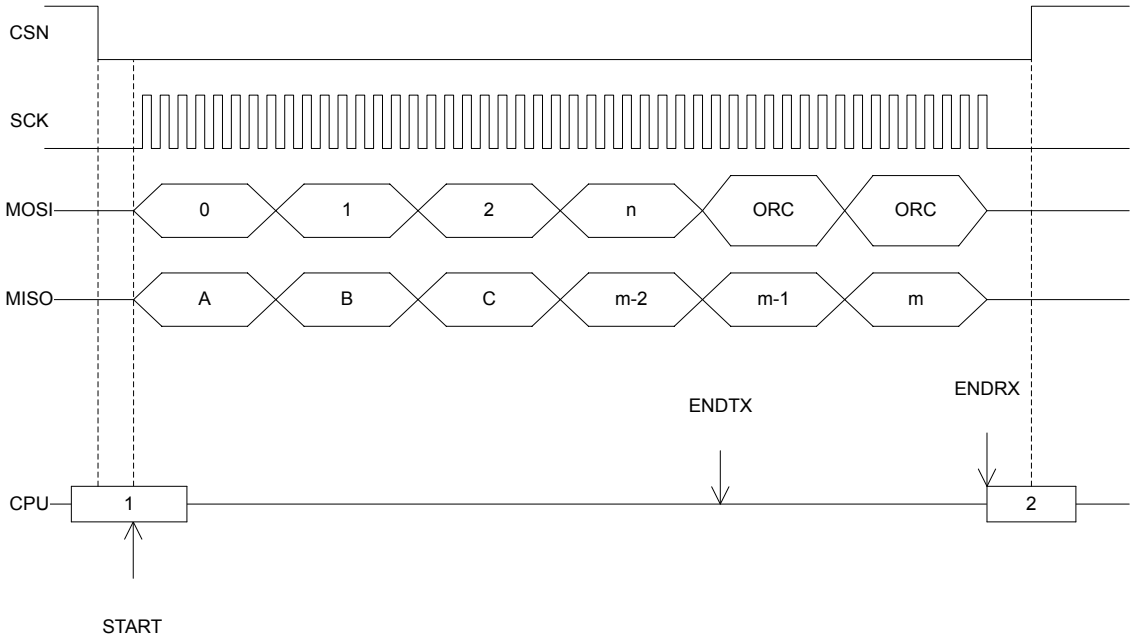

Figure 70: SPI master transaction

6.15.2 Master mode pin configuration

The SCK, MOSI, and MISO signals associated with the SPI master are mapped to physical pins according to the configuration specified in the PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively.

The PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI master is enabled, and retained only as long as the device is in ON mode. PSEL.SCK, PSEL.MOSI and PSEL.MISO must only be configured when the SPI master is disabled.

To secure correct behavior in the SPI, the pins used by the SPI must be configured in the GPIO peripheral as described in GPIO configuration on page 224 prior to enabling the SPI. This configuration must be retained in the GPIO for the selected IOs as long as the SPI is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI master signal	SPI master pin	Direction	Output value
SCK	As specified in PSEL.SCK	Output	Same as CONFIG.CPOL
MOSI	As specified in PSEL.MOSI	Output	0
MISO	As specified in PSEL.MISO	Input	Not applicable

Table 68: GPIO configuration

6.15.3 EasyDMA

The SPI master implements EasyDMA for reading and writing of data packets from and to the DATA RAM without CPU involvement.

RXD.PTR and TXD.PTR point to the RXD buffer (receive buffer) and TXD buffer (transmit buffer) respectively, see SPIM — SPI master with EasyDMA on page 223. RXD.MAXCNT and TXD.MAXCNT specify the maximum number of bytes allocated to the buffers. The SPI master will automatically stop transmitting after TXD.MAXCNT bytes have been transmitted and RXD.MAXCNT bytes have been received. If TXD.MAXCNT is larger than RXD.MAXCNT, the superfluous received bytes will be ignored. If RXD.MAXCNT is larger than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register.

If RXD.PTR and TXD.PTR are not pointing to Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next transmission immediately after having received the STARTED event.

The ENDRX/ENDTX events indicate that EasyDMA has finished accessing the RX/TX buffer in RAM respectively. The END events are generated when both RX and TX are finished accessing the buffers in RAM.

EasyDMA supports the following list types:

- Array list

6.15.3.1 EasyDMA array list

The EasyDMA array list can be represented by the data structure ArrayList_type.

For illustration, see the code example below. This data structure includes only a buffer with size equal to Channel.MAXCNT. EasyDMA will use the Channel.MAXCNT register to determine when the buffer is full. Replace 'Channel' by the specific data channel you want to use, for instance 'NRF_SPIM->RXD', 'NRF_SPIM->TXD', 'NRF_TWIM->RXD', etc.

The Channel.MAXCNT register cannot be specified larger than the actual size of the buffer. If Channel.MAXCNT is specified larger than the size of the buffer, the EasyDMA channel may overflow the buffer.

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

```
#define BUFFER_SIZE  4

typedef struct ArrayList
{
  uint8_t buffer[BUFFER_SIZE];
} ArrayList_type;

ArrayList_type MyArrayList[3];

//replace 'Channel' below by the specific data channel you want to use,
//         for instance 'NRF_SPIM->RXD', 'NRF_TWIM->RXD', etc.
Channel.MAXCNT = BUFFER_SIZE;
Channel.PTR = &MyArrayList;
```

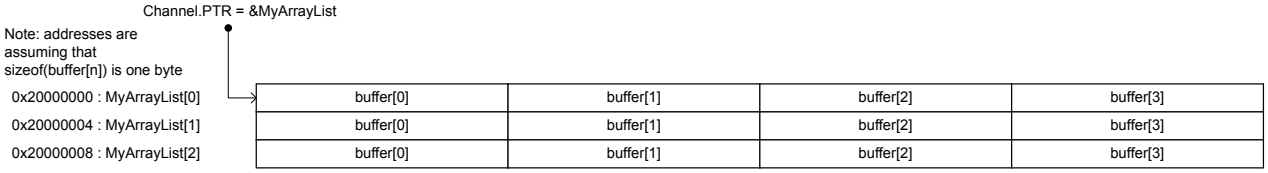

Figure 71: EasyDMA array list

6.15.4 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.15.5 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	SPIM	SPIM0 : S SPIM0 : NS	US	SA	SPI master 0	
0x50009000 0x40009000	SPIM	SPIM1 : S SPIM1 : NS	US	SA	SPI master 1	
0x5000A000 0x4000A000	SPIM	SPIM2 : S SPIM2 : NS	US	SA	SPI master 2	
0x5000B000 0x4000B000	SPIM	SPIM3 : S SPIM3 : NS	US	SA	SPI master 3	

Table 69: Instances

Register	Offset	Security	Description
TASKS_START	0x010		Start SPI transaction
TASKS_STOP	0x014		Stop SPI transaction
TASKS_SUSPEND	0x01C		Suspend SPI transaction
TASKS_RESUME	0x020		Resume SPI transaction
SUBSCRIBE_START	0x090		Subscribe configuration for task START
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
EVENTS_STOPPED	0x104		SPI transaction has stopped
EVENTS_ENDRX	0x110		End of RXD buffer reached
EVENTS_END	0x118		End of RXD buffer and TXD buffer reached
EVENTS_ENDTX	0x120		End of TXD buffer reached
EVENTS_STARTED	0x14C		Transaction started
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_END	0x198		Publish configuration for event END
PUBLISH_ENDTX	0x1A0		Publish configuration for event ENDTX
PUBLISH_STARTED	0x1CC		Publish configuration for event STARTED
SHORTS	0x200		Shortcuts between local events and tasks
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ENABLE	0x500		Enable SPIM
PSEL.SCK	0x508		Pin select for SCK
PSEL.MOSI	0x50C		Pin select for MOSI signal
PSEL.MISO	0x510		Pin select for MISO signal
FREQUENCY	0x524		SPI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
RXD.LIST	0x540		EasyDMA list type
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
TXD.LIST	0x550		EasyDMA list type
CONFIG	0x554		Configuration register
ORC	0x5C0		Over-read character. Character clocked out in case and over-read of the TXD buffer.

Table 70: Register overview

6.15.5.1 TASKS_START

Address offset: 0x010

Start SPI transaction

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_START			Start SPI transaction
			Trigger	1	Trigger task

6.15.5.2 TASKS_STOP

Address offset: 0x014

Stop SPI transaction

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

6.15.5.3 TASKS_SUSPEND

Address offset: 0x01C

Suspend SPI transaction

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend SPI transaction
			Trigger	1	Trigger task

6.15.5.4 TASKS_RESUME

Address offset: 0x020

Resume SPI transaction

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume SPI transaction
			Trigger	1	Trigger task

6.15.5.5 SUBSCRIBE_START

Address offset: 0x090

Subscribe configuration for task START

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

6.15.5.6 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

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

6.15.5.7 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

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

6.15.5.8 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

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

6.15.5.9 EVENTS_STOPPED

Address offset: 0x104

SPI transaction has stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			SPI transaction has stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.15.5.10 EVENTS_ENDRX

Address offset: 0x110

End of RXD buffer reached

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			End of RXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.15.5.11 EVENTS_END

Address offset: 0x118

End of RXD buffer and TXD buffer reached

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

6.15.5.12 EVENTS_ENDTX

Address offset: 0x120

End of TXD buffer reached

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			End of TXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.15.5.13 EVENTS_STARTED

Address offset: 0x14C

Transaction started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STARTED			Transaction started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.15.5.14 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

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

6.15.5.15 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

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

6.15.5.16 PUBLISH_END

Address offset: 0x198

Publish configuration for event END

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

6.15.5.17 PUBLISH_ENDTX

Address offset: 0x1A0

Publish configuration for event ENDTX

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

6.15.5.18 PUBLISH_STARTED

Address offset: 0x1CC

Publish configuration for event STARTED

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

6.15.5.19 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	END_START			Shortcut between event END and task START
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.15.5.20 INTENSET

Address offset: 0x304

Enable interrupt

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

6.15.5.21 INTENCLR

Address offset: 0x308

Disable interrupt

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

6.15.5.22 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	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPIM
			Disabled	0	Disable SPIM
			Enabled	7	Enable SPIM

6.15.5.23 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.15.5.24 PSEL.MOSI

Address offset: 0x50C

Pin select for MOSI signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.15.5.25 PSEL.MISO

Address offset: 0x510

Pin select for MISO signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.15.5.26 FREQUENCY

Address offset: 0x524

SPI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	FREQUENCY			SPI master data rate
			K125	0x02000000	125 kbps
			K250	0x04000000	250 kbps
			K500	0x08000000	500 kbps
			M1	0x10000000	1 Mbps
			M2	0x20000000	2 Mbps
			M4	0x40000000	4 Mbps
			M8	0x80000000	8 Mbps

6.15.5.27 RXD.PTR

Address offset: 0x534

Data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.15.5.28 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.15.5.29 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.15.5.30 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.15.5.31 TXD.PTR

Address offset: 0x544

Data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.15.5.32 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.15.5.33 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.15.5.34 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.15.5.35 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	Acce	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

6.15.5.36 ORC

Address offset: 0x5C0

Over-read character. Character clocked out in case and over-read of the TXD buffer.

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

ID	Acce	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character clocked out in case and over-read of the TXD buffer.

6.15.6 Electrical specification

6.15.6.1 SPIM master interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIM}	Bit rates for SPIM[10]	..	..	..	Mbps
$t_{SPIM,START}$	Time from START task to transmission started		1		µs

6.15.6.2 Serial Peripheral Interface Master (SPIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIM,CSCK}$	SCK period		125		ns
$t_{SPIM,RSCK,LD}$	SCK rise time, standard drive[a]			$t_{RF,25pF}$	
$t_{SPIM,RSCK,HD}$	SCK rise time, high drive[a]			$t_{HRF,25pF}$	
$t_{SPIM,FSCK,LD}$	SCK fall time, standard drive[a]			$t_{RF,25pF}$	
$t_{SPIM,FSCK,HD}$	SCK fall time, high drive[a]			$t_{HRF,25pF}$	
$t_{SPIM,WHSCK}$	SCK high time[a]	$(0.5*t_{CSCK}) - t_{RSCK}$			
$t_{SPIM,WLSCK}$	SCK low time[a]	$(0.5*t_{CSCK}) - t_{FSCK}$			
$t_{SPIM,SUMI}$	MISO to CLK edge setup time	19			ns
$t_{SPIM,HMI}$	CLK edge to MISO hold time	18			ns
$t_{SPIM,VMO}$	CLK edge to MOSI valid			59	ns
$t_{SPIM,HMO}$	MOSI hold time after CLK edge	20			ns

[10] High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

[a] At 25pF load, including GPIO pin capacitance, see GPIO spec.

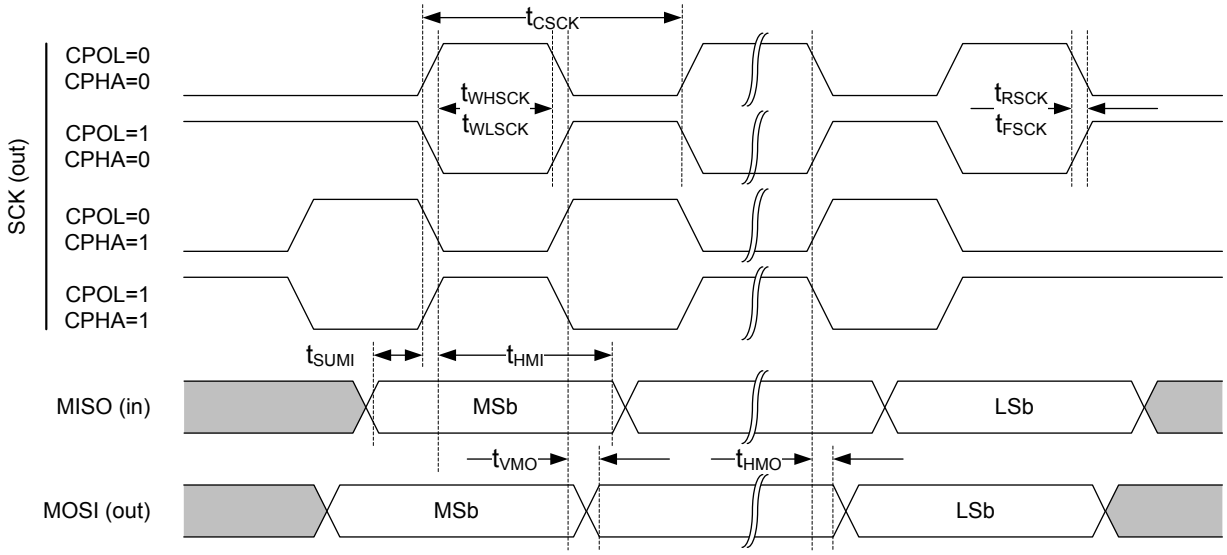

Figure 72: SPIM timing diagram

6.16 SPIS — Serial peripheral interface slave with EasyDMA

SPI slave (SPIS) is implemented with EasyDMA support for ultra low power serial communication from an external SPI master. EasyDMA in conjunction with hardware-based semaphore mechanisms removes all real-time requirements associated with controlling the SPI slave from a low priority CPU execution context.

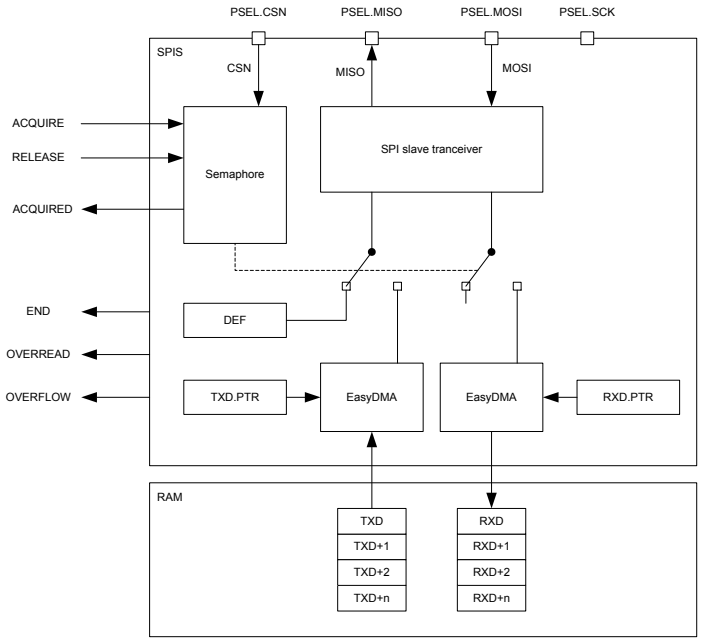

Figure 73: SPI slave

The SPIS supports SPI modes 0 through 3. The CONFIG register allows setting CPOL and CPHA appropriately.

Mode	Clock polarity	Clock phase
	CPOL	CPHA
SPI_MODE0	0 (Leading)	0 (Active High)
SPI_MODE1	0 (Leading)	1 (Active Low)
SPI_MODE2	1 (Trailing)	0 (Active High)
SPI_MODE3	1 (Trailing)	1 (Active Low)

Table 71: SPI modes

6.16.1 Shared resources

The SPI slave shares registers and other resources with other peripherals that have the same ID as the SPI slave. Therefore, you must disable all peripherals that have the same ID as the SPI slave before the SPI slave can be configured and used.

Disabling a peripheral that has the same ID as the SPI slave will not reset any of the registers that are shared with the SPI slave. It is important to configure all relevant SPI slave registers explicitly to secure that it operates correctly.

The Instantiation table in Instantiation on page 23 shows which peripherals have the same ID as the SPI slave.

6.16.2 EasyDMA

The SPI slave implements EasyDMA for reading and writing to and from the RAM. The END event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

6.16.3 SPI slave operation

SPI slave uses two memory pointers, RXD.PTR and TXD.PTR, that point to the RXD buffer (receive buffer) and TXD buffer (transmit buffer) respectively. Since these buffers are located in RAM, which can be accessed by both the SPI slave and the CPU, a hardware based semaphore mechanism is implemented to enable safe sharing.

See SPI transaction when shortcut between END and ACQUIRE is enabled on page 242.

Before the CPU can safely update the RXD.PTR and TXD.PTR pointers it must first acquire the SPI semaphore. The CPU can acquire the semaphore by triggering the ACQUIRE task and then receiving the ACQUIRED event. When the CPU has updated the RXD.PTR and TXD.PTR pointers the CPU must release the semaphore before the SPI slave will be able to acquire it. The CPU releases the semaphore by triggering the RELEASE task. This is illustrated in SPI transaction when shortcut between END and ACQUIRE is enabled on page 242. Triggering the RELEASE task when the semaphore is not granted to the CPU will have no effect.

The semaphore mechanism does not, at any time, prevent the CPU from performing read or write access to the RXD.PTR register, the TXD.PTR registers, or the RAM that these pointers are pointing to. The semaphore is only telling when these can be updated by the CPU so that safe sharing is achieved.

The semaphore is by default assigned to the CPU after the SPI slave is enabled. No ACQUIRED event will be generated for this initial semaphore handover. An ACQUIRED event will be generated immediately if the ACQUIRE task is triggered while the semaphore is assigned to the CPU.

The SPI slave will try to acquire the semaphore when CSN goes low. If the SPI slave does not manage to acquire the semaphore at this point, the transaction will be ignored. This means that all incoming data on MOSI will be discarded, and the DEF (default) character will be clocked out on the MISO line throughout the whole transaction. This will also be the case even if the semaphore is released by the CPU during the

transaction. In case of a race condition where the CPU and the SPI slave try to acquire the semaphore at the same time, as illustrated in lifeline item 2 in SPI transaction when shortcut between END and ACQUIRE is enabled on page 242, the semaphore will be granted to the CPU.

If the SPI slave acquires the semaphore, the transaction will be granted. The incoming data on MOSI will be stored in the RXD buffer and the data in the TXD buffer will be clocked out on MISO.

When a granted transaction is completed and CSN goes high, the SPI slave will automatically release the semaphore and generate the END event.

As long as the semaphore is available the SPI slave can be granted multiple transactions one after the other. If the CPU is not able to reconfigure the TXD.PTR and RXD.PTR between granted transactions, the same TX data will be clocked out and the RX buffers will be overwritten. To prevent this from happening, the END_ACQUIRE shortcut can be used. With this shortcut enabled the semaphore will be handed over to the CPU automatically after the granted transaction has completed, giving the CPU the ability to update the TXPTR and RXPTR between every granted transaction.

If the CPU tries to acquire the semaphore while it is assigned to the SPI slave, an immediate handover will not be granted. However, the semaphore will be handed over to the CPU as soon as the SPI slave has released the semaphore after the granted transaction is completed. If the END_ACQUIRE shortcut is enabled and the CPU has triggered the ACQUIRE task during a granted transaction, only one ACQUIRE request will be served following the END event.

The MAXRX register specifies the maximum number of bytes the SPI slave can receive in one granted transaction. If the SPI slave receives more than MAXRX number of bytes, an OVERFLOW will be indicated in the STATUS register and the incoming bytes will be discarded.

The MAXTX parameter specifies the maximum number of bytes the SPI slave can transmit in one granted transaction. If the SPI slave is forced to transmit more than MAXTX number of bytes, an OVERREAD will be indicated in the STATUS register and the ORC character will be clocked out.

The RXD.AMOUNT and TXD.AMOUNT registers are updated when a granted transaction is completed. The TXD.AMOUNT register indicates how many bytes were read from the TX buffer in the last transaction, that is, ORC (over-read) characters are not included in this number. Similarly, the RXD.AMOUNT register indicates how many bytes were written into the RX buffer in the last transaction.

The ENDRX event is generated when the RX buffer has been filled.

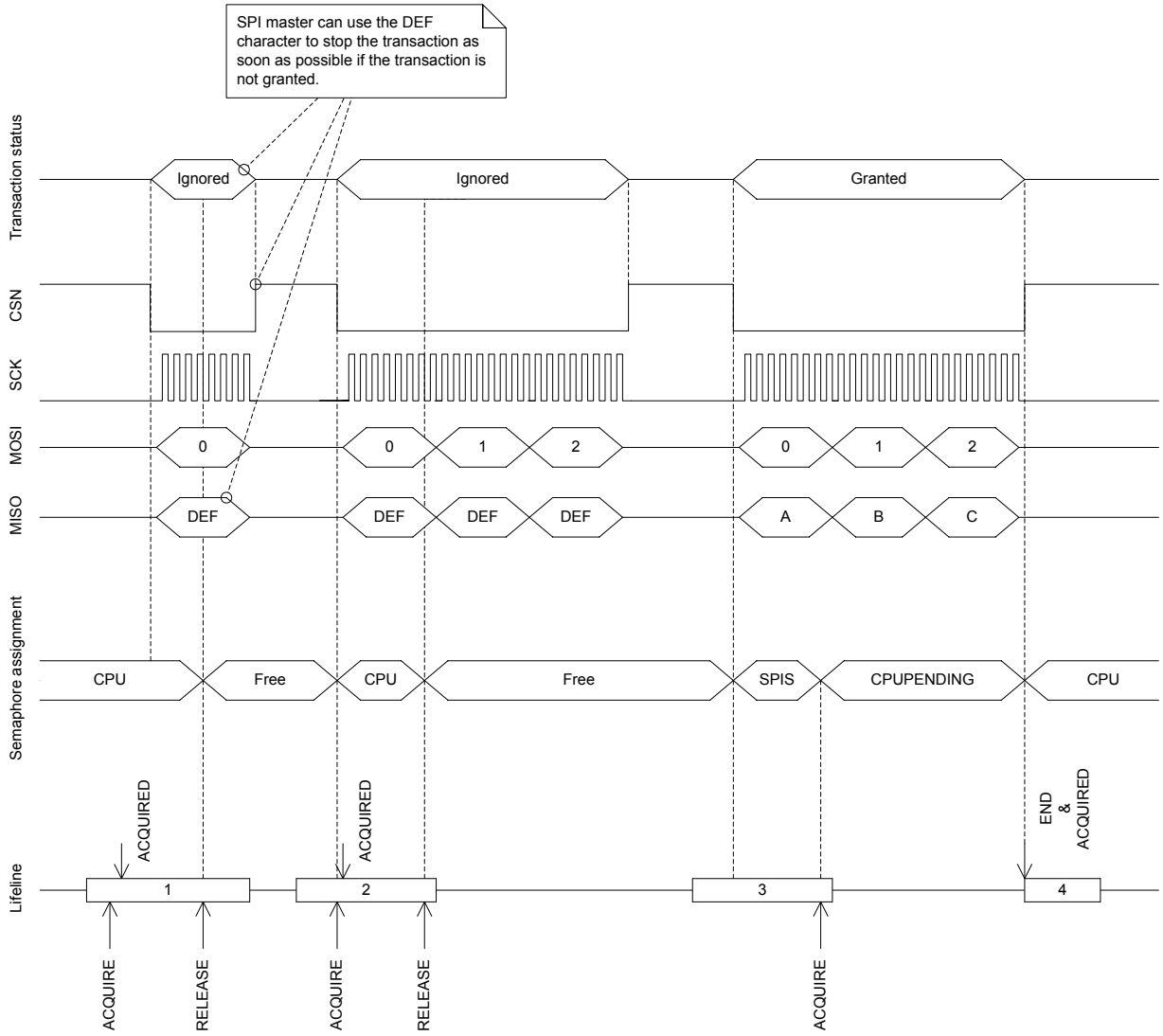

Figure 74: SPI transaction when shortcut between END and ACQUIRE is enabled

6.16.4 Pin configuration

The CSN, SCK, MOSI, and MISO signals associated with the SPI slave are mapped to physical pins according to the configuration specified in the PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers respectively. If the CONNECT field of any of these registers is set to Disconnected, the associated SPI slave signal will not be connected to any physical pins.

The PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO registers and their configurations are only used as long as the SPI slave is enabled, and retained only as long as the device is in System ON mode, see POWER — Power control on page 58 chapter for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.CSN, PSEL.SCK, PSEL.MOSI, and PSEL.MISO must only be configured when the SPI slave is disabled.

To secure correct behavior in the SPI slave, the pins used by the SPI slave must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 243 before enabling the SPI slave. This is to secure that the pins used by the SPI slave are driven correctly if the SPI slave itself is temporarily disabled, or if the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the SPI slave is to be recognized by an external SPI master.

The MISO line is set in high impedance as long as the SPI slave is not selected with CSN.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

SPI signal	SPI pin	Direction	Output value	Comment
CSN	As specified in PSEL.CSN	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
MOSI	As specified in PSEL.MOSI	Input	Not applicable	
MISO	As specified in PSEL.MISO	Input	Not applicable	Emulates that the SPI slave is not selected.

Table 72: GPIO configuration before enabling peripheral

6.16.5 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	SPIS	SPIS0 : S SPIS0 : NS	US	SA	SPI slave 0	
0x50009000 0x40009000	SPIS	SPIS1 : S SPIS1 : NS	US	SA	SPI slave 1	
0x5000A000 0x4000A000	SPIS	SPIS2 : S SPIS2 : NS	US	SA	SPI slave 2	
0x5000B000 0x4000B000	SPIS	SPIS3 : S SPIS3 : NS	US	SA	SPI slave 3	

Table 73: Instances

Register	Offset	Security	Description	
TASKS_ACQUIRE	0x024		Acquire SPI semaphore	
TASKS_RELEASE	0x028		Release SPI semaphore, enabling the SPI slave to acquire it	
SUBSCRIBE_ACQUIRE	0x0A4		Subscribe configuration for task ACQUIRE	
SUBSCRIBE_RELEASE	0x0A8		Subscribe configuration for task RELEASE	
EVENTS_END	0x104		Granted transaction completed	
EVENTS_ENDRX	0x110		End of RXD buffer reached	
EVENTS_ACQUIRED	0x128		Semaphore acquired	
PUBLISH_END	0x184		Publish configuration for event END	
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX	
PUBLISH_ACQUIRED	0x1A8		Publish configuration for event ACQUIRED	
SHORTS	0x200		Shortcuts between local events and tasks	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
SEMSTAT	0x400		Semaphore status register	
STATUS	0x440		Status from last transaction	
ENABLE	0x500		Enable SPI slave	
PSEL.SCK	0x508		Pin select for SCK	
PSEL.MISO	0x50C		Pin select for MISO signal	
PSEL.MOSI	0x510		Pin select for MOSI signal	
PSEL.CSN	0x514		Pin select for CSN signal	
PSELSCK	0x508		Pin select for SCK	Deprecated
PSELMISO	0x50C		Pin select for MISO	Deprecated
PSELMOSI	0x510		Pin select for MOSI	Deprecated
PSELCSN	0x514		Pin select for CSN	Deprecated
RXD.PTR	0x534		RXD data pointer	
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer	
RXD.AMOUNT	0x53C		Number of bytes received in last granted transaction	

Register	Offset	Security	Description	
RXDPTR	0x534		RXD data pointer	Deprecated
MAXRX	0x538		Maximum number of bytes in receive buffer	Deprecated
AMOUNTRX	0x53C		Number of bytes received in last granted transaction	Deprecated
TXD.PTR	0x544		TXD data pointer	
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer	
TXD.AMOUNT	0x54C		Number of bytes transmitted in last granted transaction	
TXDPTR	0x544		TXD data pointer	Deprecated
MAXTX	0x548		Maximum number of bytes in transmit buffer	Deprecated
AMOUNTTX	0x54C		Number of bytes transmitted in last granted transaction	Deprecated
CONFIG	0x554		Configuration register	
DEF	0x55C		Default character. Character clocked out in case of an ignored transaction.	
ORC	0x5C0		Over-read character	

Table 74: Register overview

6.16.5.1 TASKS_ACQUIRE

Address offset: 0x024

Acquire SPI semaphore

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

Value	Description
	Acquire SPI semaphore
1	Trigger task

6.16.5.2 TASKS_RELEASE

Address offset: 0x028

Release SPI semaphore, enabling the SPI slave to acquire it

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

Value	Description
	Release SPI semaphore, enabling the SPI slave to acquire it
1	Trigger task

6.16.5.3 SUBSCRIBE_ACQUIRE

Address offset: 0x0A4

Subscribe configuration for task ACQUIRE

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

6.16.5.4 SUBSCRIBE_RELEASE

Address offset: 0x0A8

Subscribe configuration for task RELEASE

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

6.16.5.5 EVENTS_END

Address offset: 0x104

Granted transaction completed

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_END			Granted transaction completed
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.16.5.6 EVENTS_ENDRX

Address offset: 0x110

End of RXD buffer reached

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			End of RXD buffer reached
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.16.5.7 EVENTS_ACQUIRED

Address offset: 0x128

Semaphore acquired

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ACQUIRED			Semaphore acquired
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.16.5.8 PUBLISH_END

Address offset: 0x184

Publish configuration for event END

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

6.16.5.9 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

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

6.16.5.10 PUBLISH_ACQUIRED

Address offset: 0x1A8

Publish configuration for event ACQUIRED

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

6.16.5.11 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	END_ACQUIRE			Shortcut between event END and task ACQUIRE
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.16.5.12 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	Acce	Field	Value ID	Value	Description
A	RW	END			Write '1' to enable interrupt for event END
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to enable interrupt for event ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	ACQUIRED			Write '1' to enable interrupt for event ACQUIRED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.16.5.13 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	Acce	Field	Value ID	Value	Description
A	RW	END			Write '1' to disable interrupt for event END
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

Bit number				31 30 29 28 27 26 25 24 23 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	Acce	Field	Value ID	Value	Description
C	RW	ACQUIRED			Write '1' to disable interrupt for event ACQUIRED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.16.5.14 SEMSTAT

Address offset: 0x400

Semaphore status register

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

6.16.5.15 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	Acce	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'

6.16.5.16 ENABLE

Address offset: 0x500

Enable SPI slave

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable SPI slave
			Disabled	0	Disable SPI slave
			Enabled	2	Enable SPI slave

6.16.5.17 PSEL.SCK

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.16.5.18 PSEL.MISO

Address offset: 0x50C

Pin select for MISO signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.16.5.19 PSEL.MOSI

Address offset: 0x510

Pin select for MOSI signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.16.5.20 PSEL.CSN

Address offset: 0x514

Pin select for CSN signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.16.5.21 PSELSCK (Deprecated)

Address offset: 0x508

Pin select for SCK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PSELSCK		[0..31]	Pin number configuration for SPI SCK signal
			Disconnected	0xFFFFFFFF	Disconnect

6.16.5.22 PSELMISO (Deprecated)

Address offset: 0x50C

Pin select for MISO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PSELMISO		[0..31]	Pin number configuration for SPI MISO signal
			Disconnected	0xFFFFFFFF	Disconnect

6.16.5.23 PSELMOSI (Deprecated)

Address offset: 0x510

Pin select for MOSI

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PSELMOSI		[0..31]	Pin number configuration for SPI MOSI signal
			Disconnected	0xFFFFFFFF	Disconnect

6.16.5.24 PSELCSN (Deprecated)

Address offset: 0x514

Pin select for CSN

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0xFFFFFFFF				**1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A	RW	PSELCSN		[0..31]	Pin number configuration for SPI CSN signal
			Disconnected	0xFFFFFFFF	Disconnect

6.16.5.25 RXD.PTR

Address offset: 0x534

RXD data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.16.5.26 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.16.5.27 RXD.AMOUNT

Address offset: 0x53C

Number of bytes received in last granted transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes received in the last granted transaction

6.16.5.28 RXDPTR (Deprecated)

Address offset: 0x534

RXD data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	RXDPTR			RXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.16.5.29 MAXRX (Deprecated)

Address offset: 0x538

Maximum number of bytes in receive buffer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXRX		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.16.5.30 AMOUNTRX (Deprecated)

Address offset: 0x53C

Number of bytes received in last granted transaction

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

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNTRX		[1..0x1FFF]	Number of bytes received in the last granted transaction

6.16.5.31 TXD.PTR

Address offset: 0x544

TXD data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			TXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.16.5.32 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.16.5.33 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transmitted in last granted transaction

6.16.5.34 TXDPTR (Deprecated)

Address offset: 0x544

TXD data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	TXDPTR			TXD data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.16.5.35 MAXTX (Deprecated)

Address offset: 0x548

Maximum number of bytes in transmit buffer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXTX		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.16.5.36 AMOUNTTX (Deprecated)

Address offset: 0x54C

Number of bytes transmitted in last granted transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNTTX		[1..0x1FFF]	Number of bytes transmitted in last granted transaction

6.16.5.37 CONFIG

Address offset: 0x554

Configuration register

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

6.16.5.38 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	Acce	Field	Value ID	Value	Description
A	RW	DEF			Default character. Character clocked out in case of an ignored transaction.

6.16.5.39 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	Acce	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character clocked out after an over-read of the transmit buffer.

6.16.6 Electrical specification

6.16.6.1 SPIS slave interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
f_{SPIS}	Bit rates for SPIS[11]			8[12]	Mbps
$t_{SPIS,START}$	Time from RELEASE task to receive/transmit (CSN active)	..	..	..	µs

6.16.6.2 Serial Peripheral Interface Slave (SPIS) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$t_{SPIS,CSCKIN}$	SCK input period	..	..	..	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			ns
$t_{SPIS,HCSN}$	CLK to CSN hold time	2000			ns
$t_{SPIS,ASA}$	CSN to MISO driven	..	..	..	ns
$t_{SPIS,ASO}$	CSN to MISO valid[a]			1000	ns
$t_{SPIS,DISSO}$	CSN to MISO disabled[a]			68	ns
$t_{SPIS,CWH}$	CSN inactive time	300			ns
$t_{SPIS,VSO}$	CLK edge to MISO valid			19	ns
$t_{SPIS,HSO}$	MISO hold time after CLK edge	18[13]			ns
$t_{SPIS,SUSI}$	MOSI to CLK edge setup time	59			ns
$t_{SPIS,HSI}$	CLK edge to MOSI hold time	20			ns

11 High bit rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

12 The actual maximum data rate depends on the master's CLK to MISO and MOSI setup and hold timings.

a At 25pF load, including GPIO capacitance, see GPIO spec.

13 This is to ensure compatibility to SPI masters sampling MISO on the same edge as MOSI is output

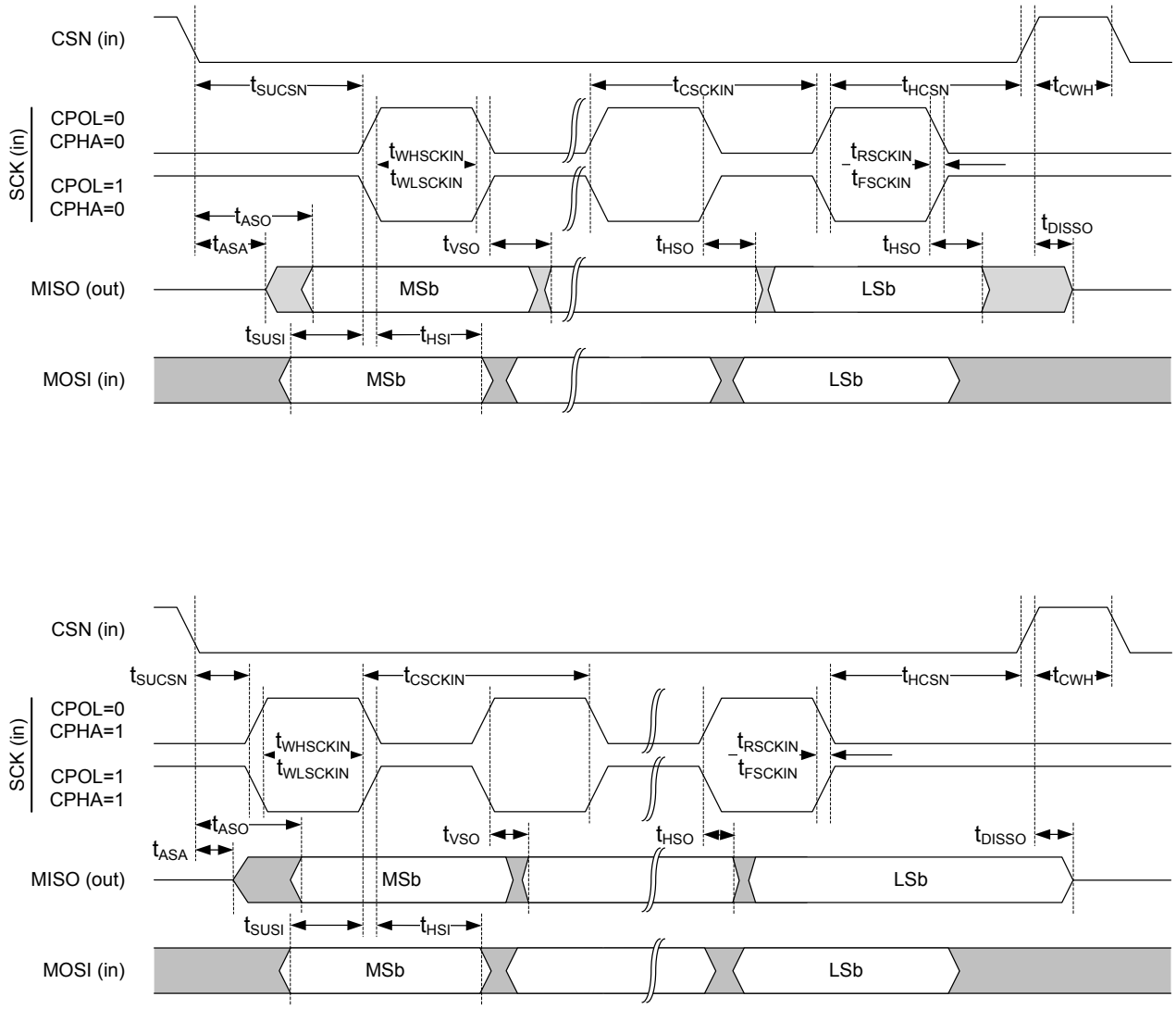

Figure 75: SPIS timing diagram

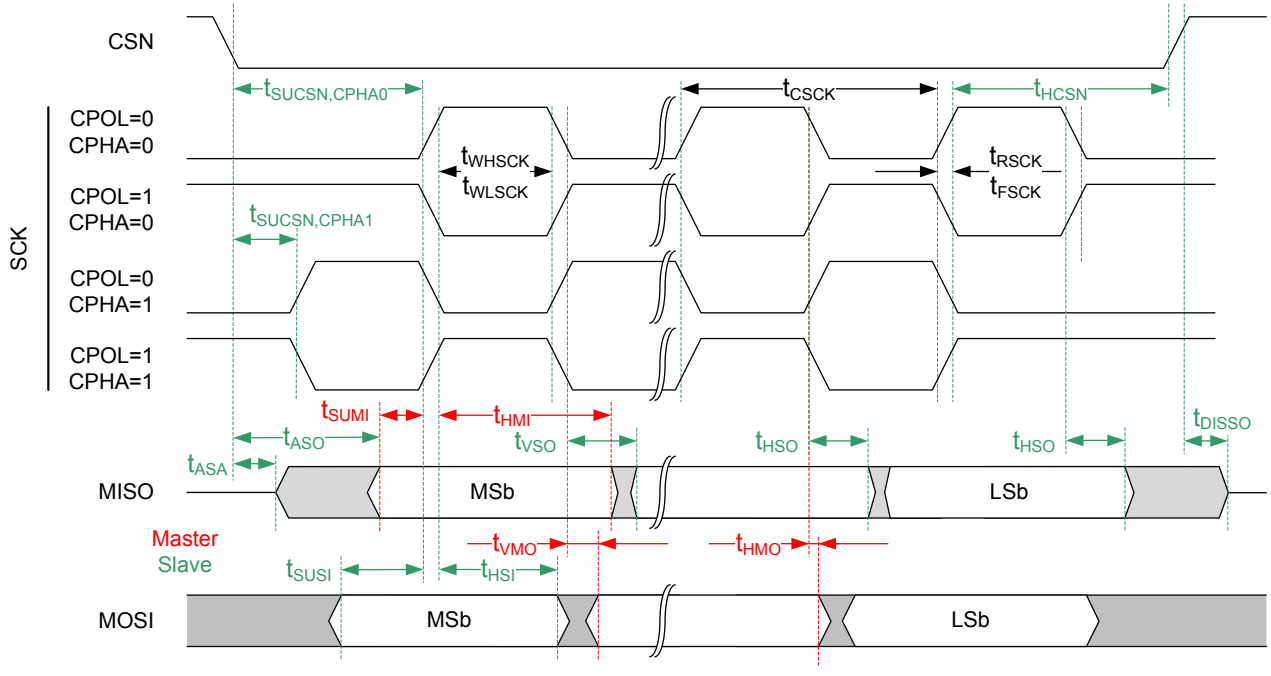

Figure 76: Common SPIM and SPIS timing diagram

6.17 SPU - System protection unit

SPU is the central point in the system to control access to memories, peripherals and other resources.

Listed here are the main features of the SPU:

- ARM® TrustZone® support, allowing definition of secure, non-secure and non-secure callable memory regions
- Extended ARM®TrustZone®, protecting memory regions and peripherals from non-CPU devices like EasyDMA transfer
- Pin access protection, preventing non-secure code and peripherals from accessing secure pin resources
- DPPI access protection, realized by preventing non-secure code and peripherals to publish from or subscribe to secured DPPI channels
- External domain access protection, controlling access rights from other MCUs

6.17.1 General concepts

SPU provides a register interface to control the various internal logic blocks that monitor access to memory-mapped slave devices (RAM, flash, peripherals, etc) and other resources (device pins, DPPI channels, etc).

For memory-mapped devices like RAM, flash and peripherals, the internal logic checks the address and attributes (e.g. read, write, execute, secure) of the incoming transfer to block it if necessary. Whether a secure resource can be accessed by a given master is defined:

For a CPU-type master

By the security state of the CPU and the security state reported by the SPU, for the address in the bus transfer

For a non-CPU master

By the security attribute of the master that initiates the transfer, defined by a SPU register

The Simplified view of the protection of RAM, flash and peripherals using SPU on page 257 shows a simplified view of the SPU registers controlling several internal modules.

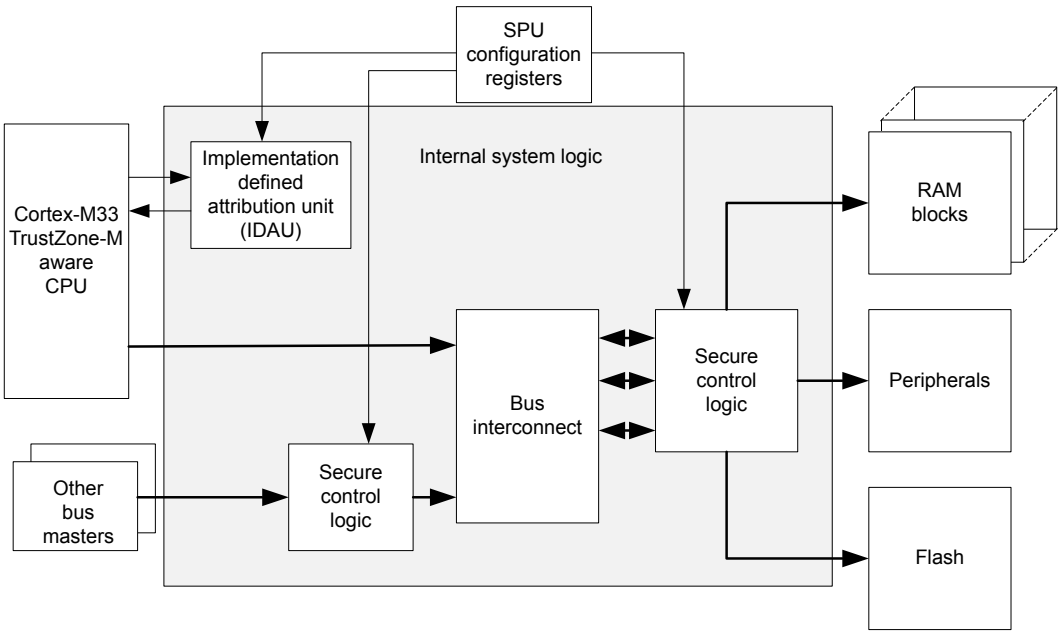

Figure 77: Simplified view of the protection of RAM, flash and peripherals using SPU

The protection logic implements a read-as-zero/write-ignore (RAZ/WI) policy:

- A blocked read operation will always return a zero value on the bus, preventing information leak
- A write operation to a forbidden region or peripheral will be ignored

An error is reported through dedicated error signals. For security state violations from an M33 master this will be a SecureFault exception, for other violations this will be an SPU event. The SPU event can be configured to generate an interrupt towards the CPU.

Other resources like pins and DPPI channels are protected by comparing the security attributes of the protected resource with the security attribute of the peripheral that wants to access it. The SPU is the only place where those security attributes can be configured.

6.17.1.1 Special considerations for ARM TrustZone for Cortex-M enabled system

For a ARM® TrustZone® for Cortex®-M enabled CPU, the SPU also controls custom logic.

Custom logic is shown as the implementation defined attribution unit (IDAU) in figure Simplified view of the protection of RAM, flash and peripherals using SPU on page 257. Full support is provided for:

- ARM® TrustZone® for Cortex®-M related instructions, like test target (TT) for reporting the security attributes of a region
- Non-secure callable (NSC) regions, to implement secure entry points from non-secure code

The SPU provides the necessary registers to configure the security attributes for memory regions and peripherals. However, as a requirement to use the SPU, the secure attribution unit (SAU) needs to be disabled and all memory set as non-secure in the ARM core. This will allow the SPU to control the IDAU and set the security attribution of all addresses as originally intended.

6.17.2 Flash access control

The flash memory space is divided into 32 regions of 32 KiB. For each region, four different types of permissions can be configured.

The four types of permissions are:

Read
Allows data read access to the region. Note that code fetch from this region is not controlled by the read permission but by the execute permission described below.

Write
Allows write or erase access to the region

Execute
Allows code fetch from this region, even if data read is disabled

Secure
Allows only bus accesses with the security attribute set to access the region

Permissions can be set independently. For example, it is possible to configure a flash region to be accessible only through secure transfer, being read-only (no write allowed) and non-executable (no code fetch allowed). For each region, permissions can be set and then locked by using the FLASHPERM[].PERM.LOCK bit, to prevent subsequent modifications.

Note that the debugger is able to step through execute-protected memory regions.

The following figure shows the flash memory space and the divided regions:

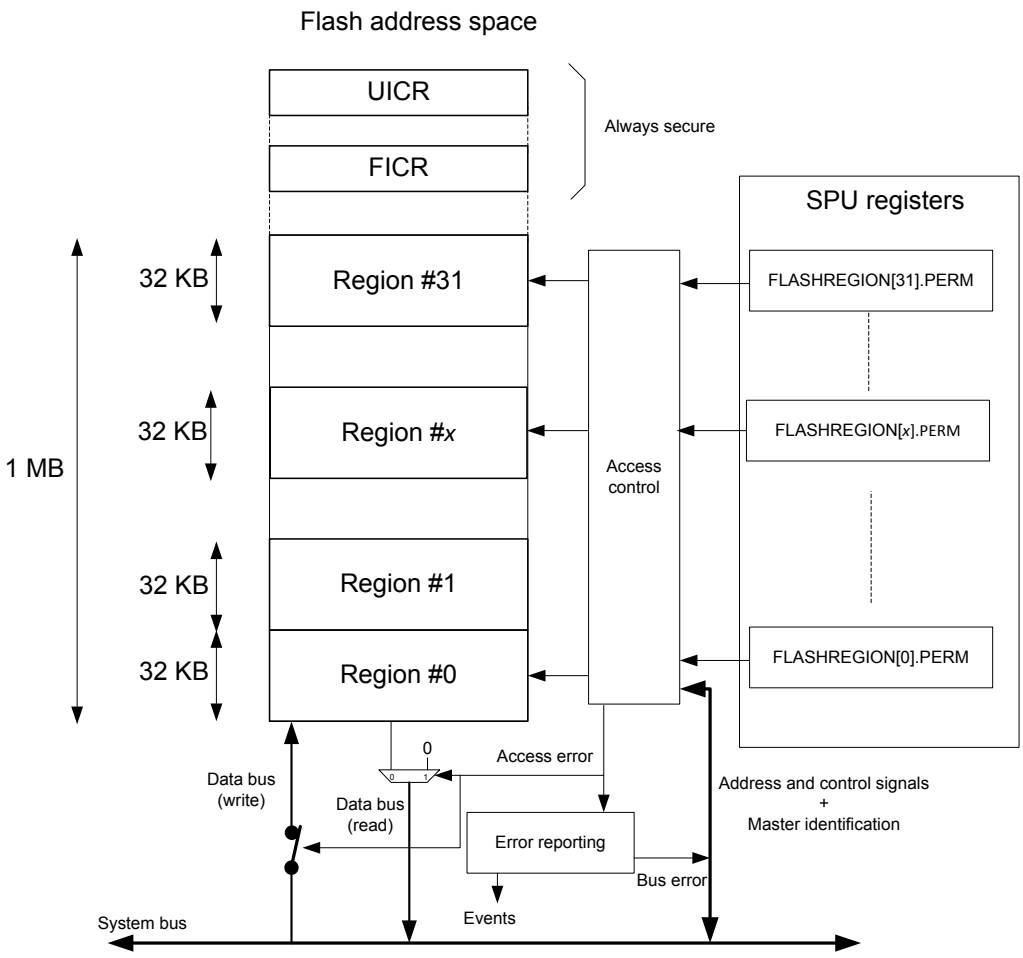

Figure 78: Region definition in the flash memory space

6.17.2.1 Non-secure callable (NSC) region definition in flash

The SPU provides support for the definition of non-secure callable (NSC) sub-regions to allow non-secure to secure function calls.

A non-secure callable sub-region can only exist within an existing secure region and its definition is done using two registers:

- FLASHNSC[].REGION, used to select the secure region that will contain the NSC sub-region
- FLASHNSC[].SIZE, used to define the size of the NSC sub-region within the secure region

The NSC sub-region will be defined from the highest address in that region, going downwards. Figure below illustrates the NSC sub-regions and the registers used for their definition:

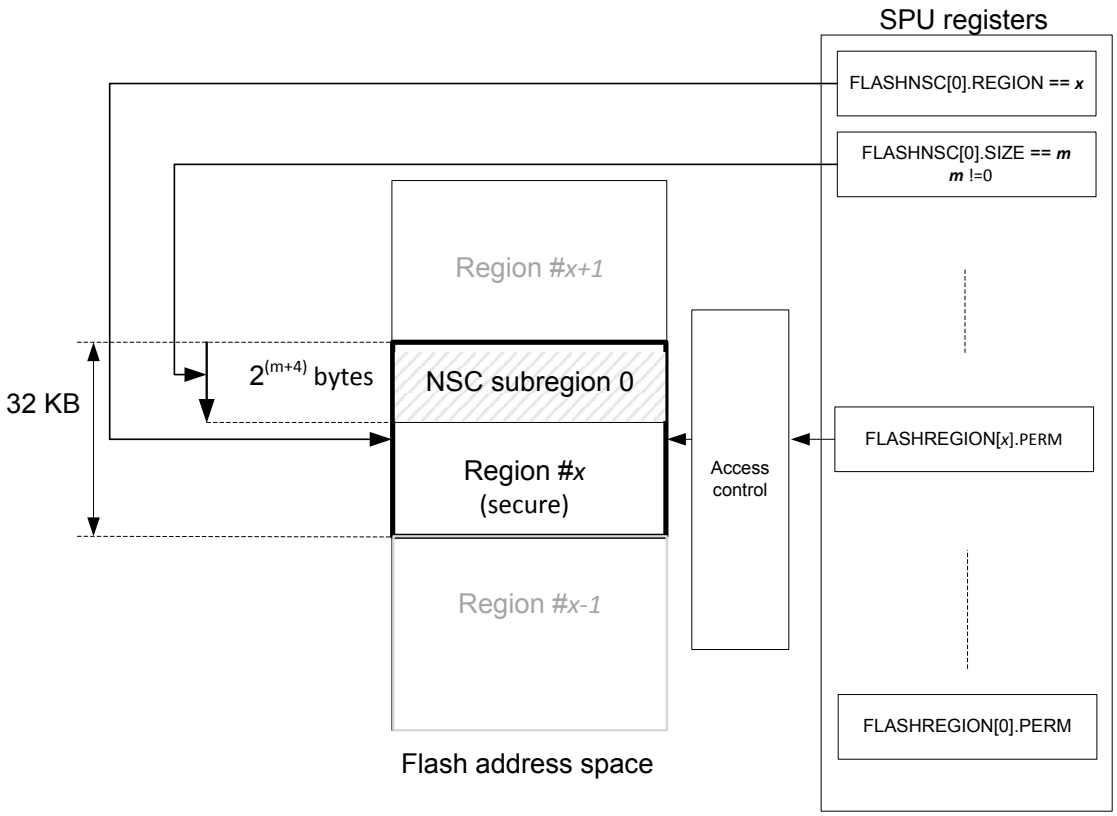

Figure 79: Non-secure callable region definition in the flash memory space

The NSC sub-region will only be defined if:

- FLASHNSC[*i*].SIZE value is non zero
- FLASHNSC[*i*].REGION defines a secure region

If FLASHNSC[*i*].REGION and FLASHNSC[*j*].REGION have the same value, there is only one sub-region defined as NSC, with the size given by the maximum of FLASHNSC[*i*].SIZE and FLASHNSC[*j*].SIZE.

If FLASHNSC[*i*].REGION defines a non-secure region, then there is no non-secure callable region defined and the selected region stays non-secure.

6.17.2.2 Flash access error reporting

The SPU and the logic controlled by it will respond with a certain behavior once an access violation is detected.

The following will happen once the logic controlled by the SPU detects an access violation on one of the flash ports:

- The faulty transfer will be blocked
- In case of a read transfer, the bus will be driven to zero
- Feedback will be sent to the master through specific bus error signals, if this is supported by the master. Moreover, the SPU will receive an event that can optionally trigger an interrupt towards the CPU.
- SecureFault exception will be triggered if security violation is detected for access from Cortex®-M33
- BusFault exception will be triggered when read/write/execute protection violation is detected for Cortex®-M33
- FLASHACCERR event will be triggered if any access violations are detected for all master types except for Cortex®-M33 security violation

The following table summarizes the SPU behavior based on the type of initiator and access violation:

Master type	Security violation	Read/Write/Execute protection violation
Cortex®-M33	SecureFault exception	BusFault exception, FLASHACCERR event
EasyDMA	RAZ/WI, FLASHACCERR event	RAZ/WI, FLASHACCERR event
Other masters	RAZ/WI, FLASHACCERR event	RAZ/WI, FLASHACCERR event

Table 75: Error reporting for flash access errors

For a Cortex®-M33 master, the SecureFault exception will take precedence over the BusFault exception if a security violation occurs simultaneously with another type of violation.

6.17.2.3 UICR and FICR protections

The user information configuration registers (UICR) and factory information configuration registers (FICR) are always considered as secure. FICR registers are read-only. UICR registers can be read and written by secure code only.

Writing new values to user information configuration registers must follow the procedure described in NVMC — Non-volatile memory controller on page 28. Code execution from FICR and UICR address spaces will always be reported as access violation, an exception to this rule applies during a debug session.

6.17.3 RAM access control

Each RAM memory space region has a set of permissions that can be set independently.

The RAM memory space is divided into 32 regions of 8 KiB.

For each region, four different types of permissions can be configured:

Read
Allows data read access to the region. Code fetch from this region is not controlled by the read permission but by the execute permission described below.

Write
Allows write access to the region

Execute
Allows code fetch from this region

Secure
Allows only bus accesses with the security attribute set to access the region

Permissions can be set independently. For example, it is possible to configure a RAM region to be accessible only through secure transfer, being read-only (no write allowed) and non-executable (no code fetch allowed). For each region, permissions can be set and then locked to prevent subsequent modifications by using the RAMPERM[].PERM.LOCK bit.

The following figure shows the RAM memory space and the devided regions:

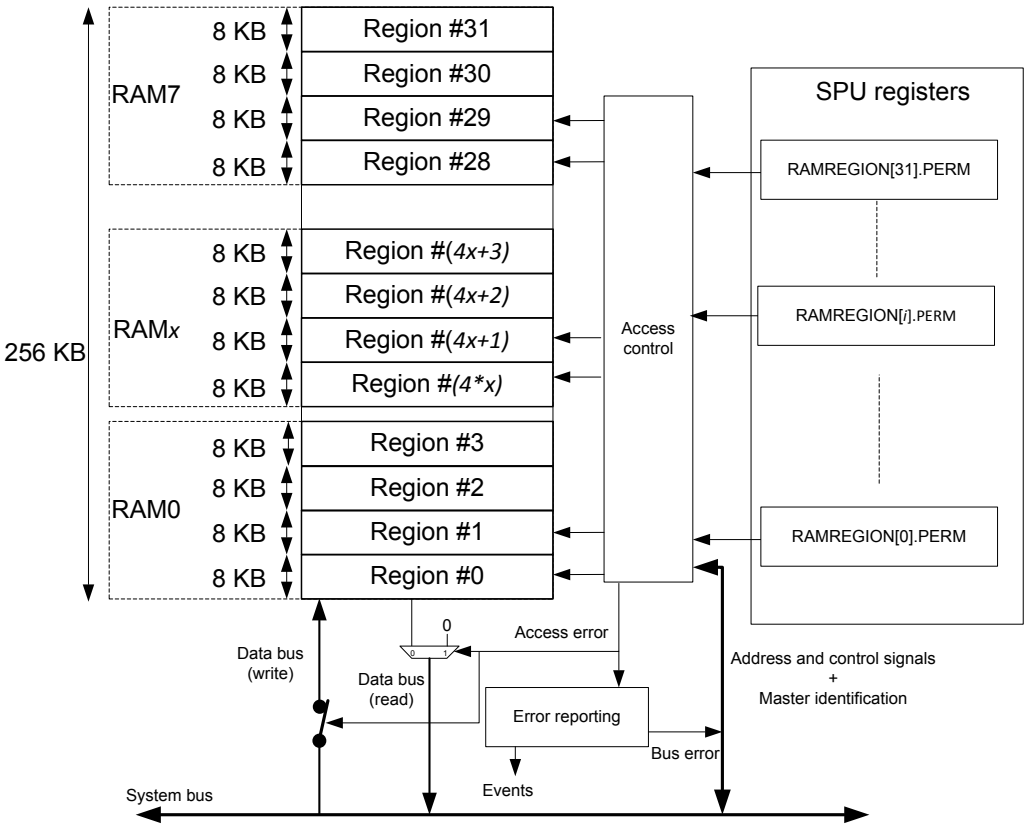

Figure 80: Region definition in the RAM memory space

6.17.3.1 Non-secure callable (NSC) region definition in RAM

The SPU provides support for the definition of non-secure callable (NSC) sub-regions to allow non-secure to secure function calls.

A non-secure callable sub-region can only exist within an existing secure region and its definition is done using two registers:

- RAMNSC[].REGION, used to select the secure region that will contain the NSC sub-region
- RAMNSC[].SIZE, used to define the size of the NSC sub-region within the secure region

The NSC sub-region will be defined from the highest address in that region, going downwards. Figure below illustrates the NSC sub-regions and the registers used for their definition:

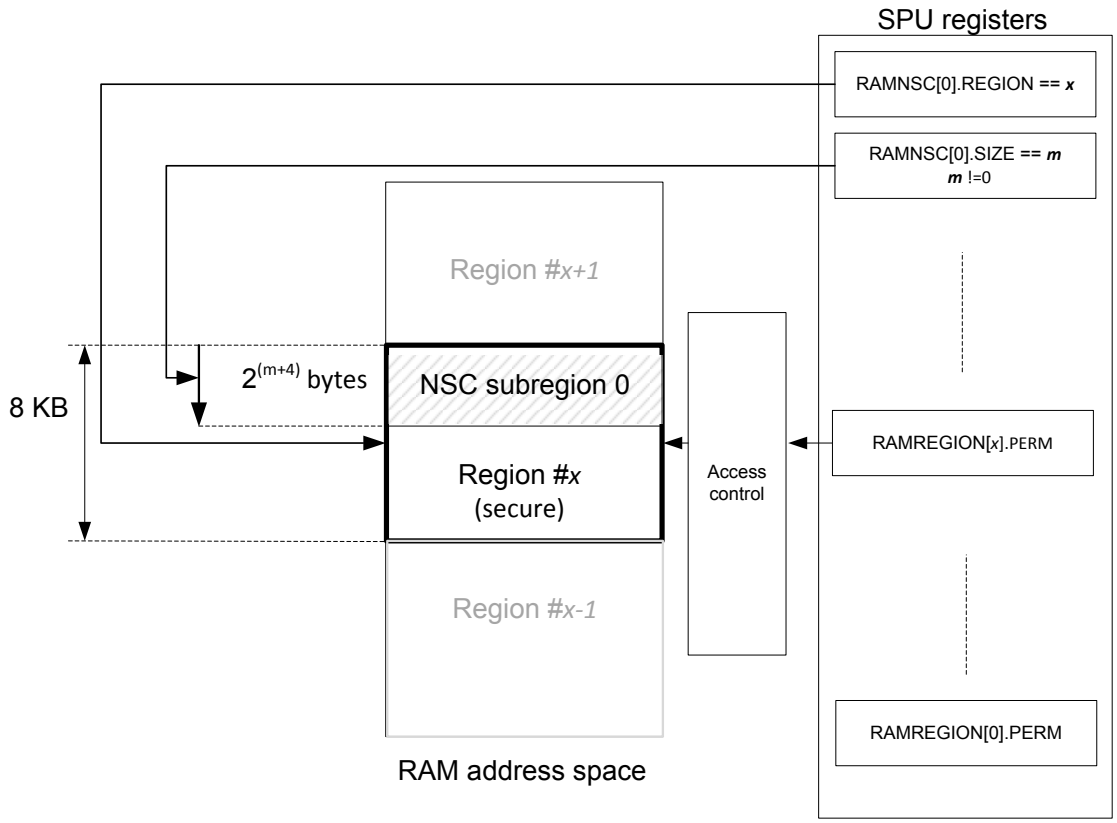

Figure 81: Non-secure callable region definition in the RAM memory space

The NSC sub-region will only be defined if:

- RAMNSC[*i*].SIZE value is non zero
- RAMNSC[*i*].REGION defines a secure region

If RAMNSC[*i*].REGION and RAMNSC[*j*].REGION have the same value, there is only one sub-region defined as NSC, with the size given by the maximum of RAMNSC[*i*].SIZE and RAMNSC[*j*].SIZE.

If RAMNSC[*i*].REGION defines a non-secure region, then there is no non-secure callable region defined and the selected region stays non-secure.

6.17.3.2 RAM access error reporting

The SPU and the logic controlled by it will respond with a certain behavior once an access violation is detected.

The following will happen once the logic controlled by the SPU detects an access violation on one of the RAM ports:

- The faulty transfer will be blocked
- In case of a read transfer, the bus will be driven to zero
- Feedback will be sent to the master through specific bus error signals, if this is supported by the master
- SecureFault exception will be triggered if security violation is detected for access from Cortex®-M33
- BusFault exception will be triggered when read/write/execute protection violation is detected for Cortex®-M33. The SPU will also generate an event that can optionally trigger an interrupt towards the CPU.
- RAMACCERR event will be triggered if any access violations are detected for all master types but for Cortex®-M33 security violation

The following table summarizes the SPU behavior based on the type of initiator and access violation:

Master type	Security violation	Read/Write/Execute protection violation
Cortex®-M33	SecureFault exception	BusFault exception, RAMACCERR event
EasyDMA	RAZ/WI, RAMACCERR event	RAZ/WI, RAMACCERR event
Other masters	RAZ/WI, RAMACCERR event	RAZ/WI, RAMACCERR event

Table 76: Error reporting for RAM access errors

For a Cortex®-M33 master, the SecureFault exception will take precedence over the BusFault exception if a security violation occurs simultaneously with another type of violation.

6.17.4 Peripheral access control

Access controls are defined by the characteristics of the peripheral.

Peripherals can have their security attribute set as:

Always secure
For a peripheral related to system control

Always non-secure
For some general-purpose peripherals

Configurable
For general-purpose peripherals that may be configured for secure only access

The full list of peripherals and their corresponding security attributes can be found in Memory map on page 22. For each peripheral with ID *id*, PERIPHID[*id*]. PERM will show whether the security attribute for this peripheral is configurable or not.

If not hardcoded, the security attribute can configured using the PERIPHID[*id*].PERM.

At reset, all user-selectable and split security peripherals are set to be secure, with secure DMA where present.

Secure code can access both secure peripherals and non-secure peripherals.

6.17.4.1 Peripherals with split security

Peripherals with split security are defined to handle use-cases when both secure and non-secure code needs to control the same resource.

When peripherals with split security have their security attribute set to non-secure, access to specific registers and bitfields within some registers is dependent on the security attribute of the bus transfer. For example, some registers will not be accessible for a non-secure transfer.

When peripherals with split security have their security attribute set to secure, then only secure transfers can access their registers.

See Instantiation on page 23 for an overview of split security peripherals. Respective peripheral chapters explain the specific security behavior of each peripheral.

6.17.4.2 Peripheral address mapping

Peripherals that have non-secure security mapping have their address starting with 0x4XXX_XXXX. Peripherals that have secure security mapping have their address starting with 0x5XXX_XXXX.

Peripherals with a user-selectable security mapping are available at an address starting with:

- 0x4XXX_XXXX, if the peripheral security attribute is set to non-secure
- 0x5XXX_XXXX, if the peripheral security attribute is set to secure

Peripherals with a split security mapping are available at an address starting with:

- 0x4XXX_XXXX for non-secure access and 0x5XXX_XXXX for secure access, if the peripheral security attribute is set to non-secure
  - Secure registers in the 0x4XXX_XXXX range are not visible for secure or non-secure code, and an attempt to access such a register will result in write-ignore, read-as-zero behavior
  - Secure code can access both non-secure and secure registers in the 0x5XXX_XXXX range
- 0x5XXX_XXXX, if the peripheral security attribute is set to secure

Any attempt to access the 0x5000_0000-0x5FFF_FFFF address range from non-secure code will be ignored and generate a SecureFault exception.

The table below illustrates the address mapping for the three type of peripherals in all possible configurations

Security-features and configuration	Is mapped at 0x4XXX_XXXX?	Is mapped at 0x5XXX_XXXX?
Secure peripheral	No	Yes
Non-secure peripheral	Yes	No
Split-security peripheral, with attribute=secure	No	Yes
Split-security peripheral, with attribute=non-secure	Yes, restricted functionality	Yes

Table 77: Peripheral's address mapping in relation to its security-features and configuration

6.17.4.3 Special considerations for peripherals with DMA master

Peripherals containing a DMA master can be configured so the security attribute of the DMA transfers is different from the security attribute of the peripheral itself. This allows a secure peripheral to do non-secure data transfers to or from the system memories.

The following conditions must be met:

- The DMA field of PERIPHID[].PERM.SECURITYMAPPING should read as "SeparateAttribute"
- The peripheral itself should be secure (PERIPHID[].PERM.SECATTR == 1)

Then it is possible to select the security attribute of the DMA transfers using the field DMASEC (PERIPHID[].PERM.DMASEC == Secure and PERIPHID[].PERM.DMASEC == NonSecure) in PERIPHID[].PERM.

6.17.4.4 Peripheral access error reporting

Peripherals send error reports once access violation is detected.

The following will happen if the logic controlled by the SPU detects an access violation on one of the peripherals:

- The faulty transfer will be blocked
- In case of a read transfer, the bus will be driven to zero
- Feedback is sent to the master through specific bus error signals, if this is supported by the master. If the master is a processor supporting ARM® TrustZone® for Cortex®-M, a SecureFault exception will be generated for security related errors.
- The PERIPHACCERR event will be triggered

6.17.5 Pin access control

Access to device pins can be controlled by the SPU. A pin can be declared as secure so that only secure peripherals or secure code can access it.

The security attribute of each pin can be individually configured in SPU's GPIOPORT[0].PERM register. When the secure attribute is set for a pin, only peripherals that have the secure attribute set will be able to read the value of the pin or change it.

Peripherals can select the pin(s) they need access to through their PSEL register(s). If a peripheral has its attribute set to non-secure, but one of its PSEL registers selects a pin with the attribute set to secure, the SPU controlled logic will ensure that the pin selection is not propagated. In addition, the pin value will always be read as zero, to prevent a non-secure peripheral from obtaining a value from a secure pin. Whereas access to other pins with attribute set as non-secure will not be blocked.

Peripherals located in other domains (other than the application domain) can access pins only if the security attribute of the domain allows access to the pins they are trying to access. That is, secure domains can access both secure and non-secure pins, whereas non-secure domains can only access non-secure pins. This is illustrated in the following figure:

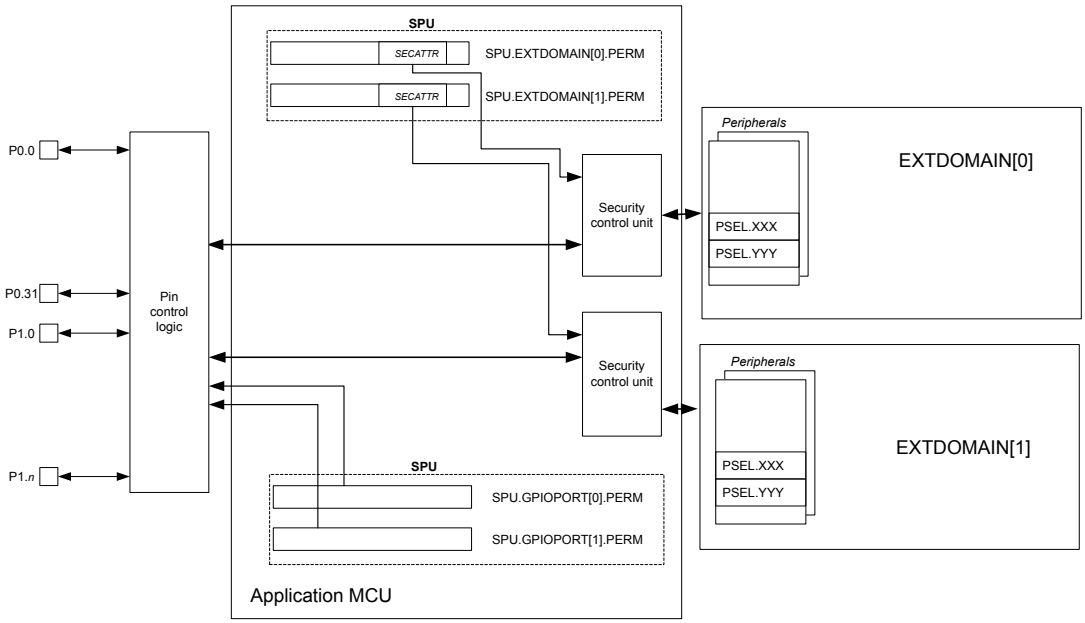

Figure 82: Pin access for domains other than the application domain

6.17.5.1 Direct pin control through the GPIO peripheral

Pins can be controlled directly through the general purpose input/output (GPIO) peripheral. It is a split-security peripheral, meaning that both secure and non-secure accesses are allowed.

Non-secure peripheral accesses will only be able to configure and control pins defined as non-secure in the GPIOPORT.PERM[] register(s). Attempts to access a register or a bitfield controlling a pin marked as secure in GPIO.PERM[] register(s) will be ignored:

- Write accesses will have no effect
- Read accesses will always return a zero value

No exception is triggered when a non-secure access targets a register or bitfield controlling a secure pin. For example, if the bit *i* is set in the GPIO.PERM[0] register (declaring Pin P0.*i* as secure), then

- non-secure write accesses to OUT, OUTSET, OUTCLR, DIR, DIRSET, DIRCLR and LATCH registers will not be able to write to bit *i* of those registers
- non-secure write accesses to register PIN_CNF[*i*] will be ignored
- non-secure read accesses to registers OUT, OUTSET, OUTCLR, IN, DIR, DIRSET, DIRCLR and LATCH will always read a 0 for the bit at position *i*
- non-secure read accesses to register PIN_CNF[*i*] will always return 0

The GPIO.DETECTMODE and GPIO.DETECTMODE_SEC registers are handled differently than the other registers mentioned before. When securely accessed, the DETECTMODE_SEC register controls the source for the DETECT_SEC signal for the pins marked as secure. Upon a non-secure access, the DETECTMODE_SEC is read as zero and write access are ignored. The GPIO.DETECTMODE register controls the source for the DETECT_NSEC signal for the pins defined as non-secured.

The DETECT_NSEC signal is routed to the non-secure GPIOTE peripheral, GPIOTE1, allowing generation of events and interrupts from pins marked as non-secured. The DETECT_SEC signal is routed to the secure GPIOTE peripheral, GPIOTE0, allowing generation of events and interrupts from pins marked as secured. The following figure illustrates how the DETECT_NSEC and DETECT_SEC signals are generated from the GPIO PIN[].DETECT signals.

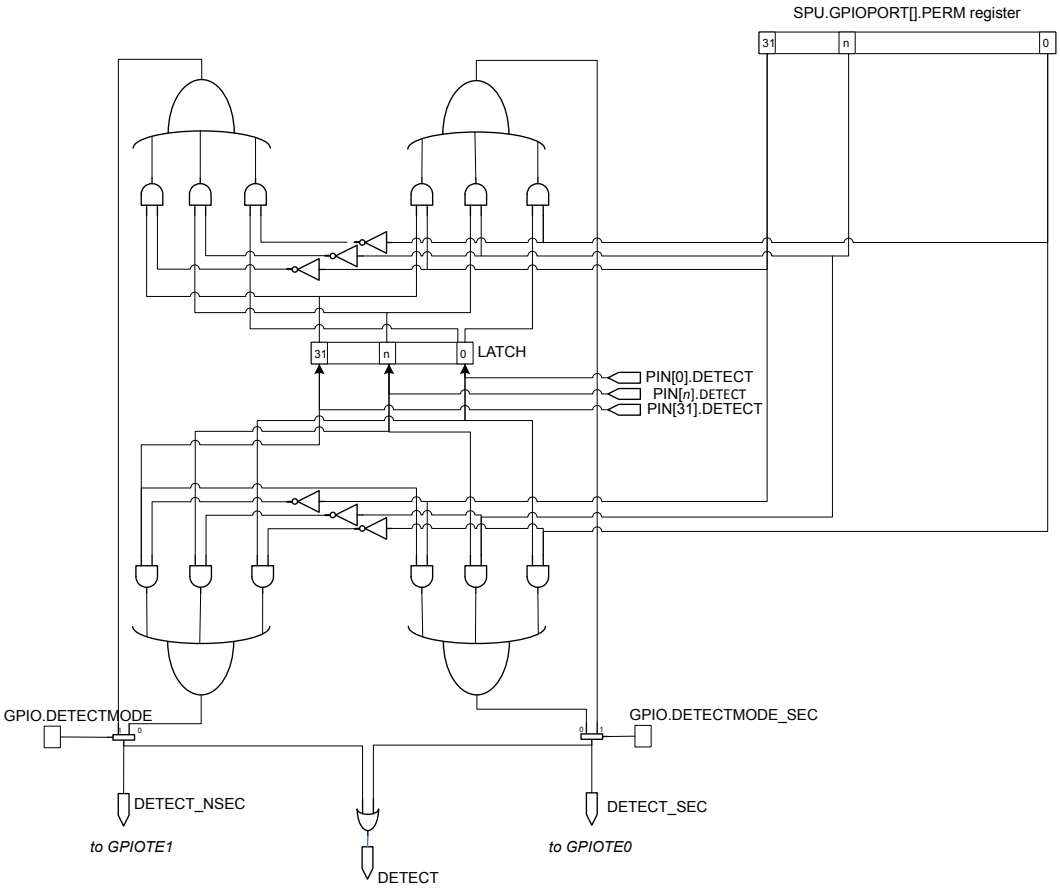

Figure 83: Principle of direct pin access

6.17.6 DPPI access control

Access to DPPI channels can be restricted. A channel can be declared as secure so that only secure peripherals can access it.

The security attribute of a DPPI channel is configured in . When the secure attribute is set for a channel, only peripherals that have the secure attribute set will be able to publish events to this channel or subscribe to this channel to receive tasks.

The DPPI controller peripheral (DPPIC) is a split security peripheral, i.e., its security behavior depends on the security attributes of both the DPPIC and the accessing party. See Special considerations regarding the DPPIC configuration registers on page 268 for more information about the DPPIC security behavior.

If a non-secure peripheral wants to publish an event on a secure DPPI channel, the channel will ignore the event. If a non-secure peripheral subscribes to a secure DPPI channel, it will not receive any events from this channel. The following figure illustrates the principle of operation of the security logic for a subscribed channel:

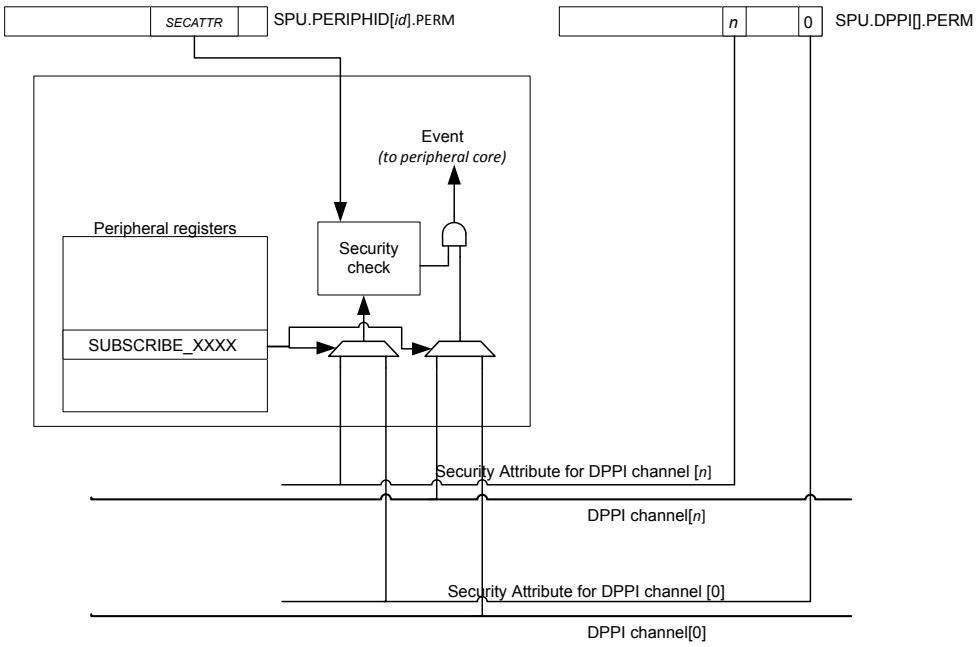

Figure 84: Subscribed channel security concept

No error reporting mechanism is associated with the DPPI access control logic.

6.17.6.1 Special considerations regarding the DPPIC configuration registers

DPPI channels can be enabled, disabled and grouped through the DPPI controller (DPPIC). The DPPIC is a split-security peripheral, and handles both secure and non-secure accesses.

A non-secure peripheral access will only be able to configure and control DPPI channels defined as non-secure in SPU's DPPI[n].PERM register(s). A secure peripheral access can control all DPPI channels, independently of the configuration in the DPPI[n].PERM register(s).

The DPPIC allows the creation of group of channels to be able to enable or disable all channels within a group simultaneously. The security attribute of a group of channels (secure or non-secure) is defined as follows:

- If all channels (enabled or not) in the group are non-secure, then the group is considered non-secure
- If at least one of the channels (enabled or not) in the group is secure, then the group is considered secure

A non-secure access to a DPPIC register, or a bitfield controlling a channel marked as secure in SPU.DDPI[n].PERM register(s), will be ignored:

- Write accesses will have no effect
- Read will always return a zero value

No exceptions are thrown when a non-secure access targets a register or bitfield controlling a secure channel. For example, if the bit *i* is set in the SPU.DPPI[0].PERM register (declaring the DPPI channel *i* as secure), then:

- Non-secure write accesses to registers CHEN, CHENSET and CHENCLR will not be able to write to bit *i* of those registers
- Non-secure write accesses to registers TASK_CHG[*j*].EN and TASK_CHG[*j*].DIS will be ignored if the channel group *j* contains at least one channel defined as secure (it can be the channel *i* itself or any channel declared as secured)

- Non-secure read accesses to registers CHEN, CHENSET and CHENCLR will always read zero for the bit at position *i*

For the channel configuration registers (DPPIC.CHG[...]), access from non-secure code is only possible if the included channels are all non-secure, whether the channels are enabled or not. If a DPPIC.CHG[*g*] register included one or more secure channels, then the group *g*is considered as secure and only a secure transfer can read or write DPPIC.CHG[*g*]. A non-secure write will be ignored and a non-secure read will return zero.

The DPPIC can subscribe to secure or non-secure channels through SUBSCRIBE_CHG[] registers in order to trigger task for enabling or disabling groups of channels. But an event from a non-secure channel will be ignored if the group subscribing to this channel is secure. An event from a secure channel can trigger both secure and non-secure tasks.

6.17.7 External domain access control

Other domains with their own CPUs can access peripherals, flash and RAM memories. The SPU allows controlling accesses from those bus masters.

The external domains can access application MCU memories and peripherals. External domains are assigned security attributes as described in register EXTDOMAIN[n].PERM.

Domain	Capability register	Permission register
LTE modem	Modem is always a non-secure domain	Not applicable

Table 78: Register mapping for external domains

The figure below illustrates how the security control units are used to assign security attributes to transfers initiated by the external domains:

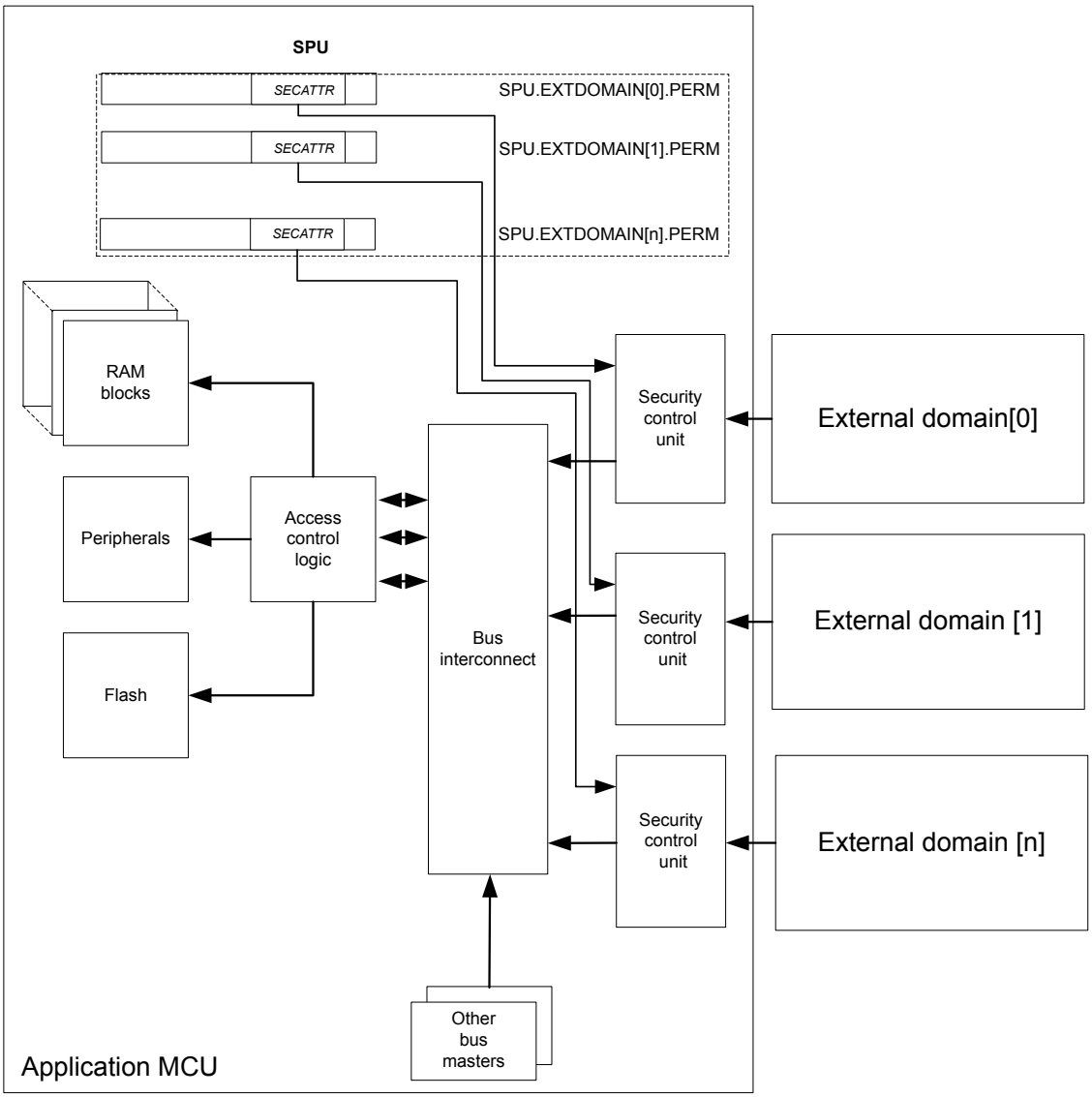

Figure 85: Access control from external domains

6.17.8 TrustZone for Cortex-M ID allocation

Flash and RAM regions, as well as non-secure and secure peripherals, are assigned unique TrustZone® IDs.

Note: TrustZone® ID should not be confounded with the peripheral ID used to identify peripherals.

The table below shows the TrustZone® ID allocation:

Regions	TrustZone Cortex-M ID
Flash regions 0..31	0..31
RAM regions 0..15	64..79
Non-secure peripherals	253
Secure peripherals	254

Table 79: TrustZone ID allocation

6.17.9 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50003000	SPU	SPU	S	NA	System Protection Unit	

Table 80: Instances

Register	Offset	Security	Description	
EVENTS_RAMACCERR	0x100		A security violation has been detected for the RAM memory space	
EVENTS_FLASHACCERR	0x104		A security violation has been detected for the flash memory space	
EVENTS_PERIPHACCERR	0x108		A security violation has been detected on one or several peripherals	
PUBLISH_RAMACCERR	0x180		Publish configuration for event RAMACCERR	
PUBLISH_FLASHACCERR	0x184		Publish configuration for event FLASHACCERR	
PUBLISH_PERIPHACCERR	0x188		Publish configuration for event PERIPHACCERR	
INTEN	0x300		Enable or disable interrupt	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
CAP	0x400		Show implemented features for the current device	
EXTDOMAIN[0].PERM	0x440		Access for bus access generated from the external domain 0 List capabilities of the external domain 0	
DPPI[0].PERM	0x480		Select between secure and non-secure attribute for the DPPI channels.	
DPPI[0].LOCK	0x484		Prevent further modification of the corresponding PERM register	
GPIOPORT[0].PERM	0x4C0		Select between secure and non-secure attribute for pins 0 to 31 of port 0.	Retained
GPIOPORT[0].LOCK	0x4C4		Prevent further modification of the corresponding PERM register	
FLASHNSC[0].REGION	0x500		Define which flash region can contain the non-secure callable (NSC) region 0	
FLASHNSC[0].SIZE	0x504		Define the size of the non-secure callable (NSC) region 0	
FLASHNSC[1].REGION	0x508		Define which flash region can contain the non-secure callable (NSC) region 1	
FLASHNSC[1].SIZE	0x50C		Define the size of the non-secure callable (NSC) region 1	
RAMNSC[0].REGION	0x540		Define which RAM region can contain the non-secure callable (NSC) region 0	
RAMNSC[0].SIZE	0x544		Define the size of the non-secure callable (NSC) region 0	
RAMNSC[1].REGION	0x548		Define which RAM region can contain the non-secure callable (NSC) region 1	
RAMNSC[1].SIZE	0x54C		Define the size of the non-secure callable (NSC) region 1	
FLASHREGION[n].PERM	0x600		Access permissions for flash region n	
RAMREGION[n].PERM	0x700		Access permissions for RAM region n	
PERIPHID[n].PERM	0x800		List capabilities and access permissions for the peripheral with ID n	

Table 81: Register overview

6.17.9.1 EVENTS_RAMACCERR

Address offset: 0x100

A security violation has been detected for the RAM memory space

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RAMACCERR			A security violation has been detected for the RAM memory space
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.17.9.2 EVENTS_FLASHACCERR

Address offset: 0x104

A security violation has been detected for the flash memory space

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_FLASHACCERR			A security violation has been detected for the flash memory space
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.17.9.3 EVENTS_PERIPHACCERR

Address offset: 0x108

A security violation has been detected on one or several peripherals

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_PERIPHACCERR			A security violation has been detected on one or several peripherals
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.17.9.4 PUBLISH_RAMACCERR

Address offset: 0x180

Publish configuration for event RAMACCERR

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

6.17.9.5 PUBLISH_FLASHACCERR

Address offset: 0x184

Publish configuration for event FLASHACCERR

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

6.17.9.6 PUBLISH_PERIPHACCERR

Address offset: 0x188

Publish configuration for event PERIPHACCERR

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

6.17.9.7 INTEN

Address offset: 0x300

Enable or disable interrupt

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

6.17.9.8 INTENSET

Address offset: 0x304

Enable interrupt

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

6.17.9.9 INTENCLR

Address offset: 0x308

Disable interrupt

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

6.17.9.10 CAP

Address offset: 0x400

Show implemented features for the current device

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000001				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1**
ID	Acce	Field	Value ID	Value	Description
A	R	TZM			Show ARM TrustZone status
			NotAvailable	0	ARM TrustZone support not available
			Enabled	1	ARM TrustZone support is available

6.17.9.11 EXTDOMAIN[n].PERM (n=0..0)

Address offset: 0x440 + (n × 0x4)

Access for bus access generated from the external domain n

List capabilities of the external domain n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	SECUREMAPPING			Define configuration capabilities for TrustZone Cortex-M secure attribute **Note:** This field is ReadOnly
			NonSecure	0	The bus access from this external domain always have the non-secure attribute set
			Secure	1	The bus access from this external domain always have the secure attribute set
			UserSelectable	2	Non-secure or secure attribute for bus access from this domain is defined by the EXTDOMAIN[n].PERM register
B	RW	SECATTR			Peripheral security mapping **Note:** This bit has effect only if EXTDOMAIN[n].PERM.SECUREMAPPING reads as UserSelectable
			NonSecure	0	Bus accesses from this domain have the non-secure attribute set
			Secure	1	Bus accesses from this domain have secure attribute set
C	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

6.17.9.12 DPPI[n].PERM (n=0..0)

Address offset: 0x480 + (n × 0x8)

Select between secure and non-secure attribute for the DPPI channels.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				P O N M L K J I H G F E D C B A
Reset 0x0000FFFF				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1**
ID	Acce	Field	Value ID	Value	Description
A-P	RW	CHANNEL[i] (i=0..15)			Select secure attribute.
			Secure	1	Channeli has its secure attribute set
			NonSecure	0	Channeli has its non-secure attribute set

6.17.9.13 DPPI[n].LOCK (n=0..0)

Address offset: 0x484 + (n × 0x8)

Prevent further modification of the corresponding PERM register

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

ID	Acce	Field	Value ID	Value	Description
A	RW	LOCK			
			Locked	1	DPPI[n].PERM register can't be changed until next reset
			Unlocked	0	DPPI[n].PERM register content can be changed

6.17.9.14 GPIOPORT[n].PERM (n=0..0) (Retained)

Address offset: 0x4C0 + (n × 0x8)

This register is a retained register

Select between secure and non-secure attribute for pins 0 to 31 of port n.

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

ID	Acce	Field	Value ID	Value	Description
A-f	RW	PIN[i] (i=0..31)			Select secure attribute attribute for PIN i.
			Secure	1	Pin i has its secure attribute set
			NonSecure	0	Pin i has its non-secure attribute set

6.17.9.15 GPIOPORT[n].LOCK (n=0..0)

Address offset: 0x4C4 + (n × 0x8)

Prevent further modification of the corresponding PERM register

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

ID	Acce	Field	Value ID	Value	Description
A	RW	LOCK			
			Locked	1	GPIOPORT[n].PERM register can't be changed until next reset
			Unlocked	0	GPIOPORT[n].PERM register content can be changed

6.17.9.16 FLASHNSC[n].REGION (n=0..1)

Address offset: 0x500 + (n × 0x8)

Define which flash region can contain the non-secure callable (NSC) region n

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

ID	Acce	Field	Value ID	Value	Description
A	RW	REGION			Region number
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

6.17.9.17 FLASHNSC[n].SIZE (n=0..1)

Address offset: 0x504 + (n × 0x8)

Define the size of the non-secure callable (NSC) region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	SIZE			Size of the non-secure callable (NSC) region n
			Disabled	0	The region n is not defined as a non-secure callable region. Normal security attributes (secure or non-secure) are enforced.
			32	1	The region n is defined as non-secure callable with a 32-byte size
			64	2	The region n is defined as non-secure callable with a 64-byte size
			128	3	The region n is defined as non-secure callable with a 128-byte size
			256	4	The region n is defined as non-secure callable with a 256-byte size
			512	5	The region n is defined as non-secure callable with a 512-byte size
			1024	6	The region n is defined as non-secure callable with a 1024-byte size
			2048	7	The region n is defined as non-secure callable with a 2048-byte size
			4096	8	The region n is defined as non-secure callable with a 4096-byte size
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

6.17.9.18 RAMNSC[n].REGION (n=0..1)

Address offset: 0x540 + (n × 0x8)

Define which RAM region can contain the non-secure callable (NSC) region n

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				B A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	REGION			Region number
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

6.17.9.19 RAMNSC[n].SIZE (n=0..1)

Address offset: 0x544 + (n × 0x8)

Define the size of the non-secure callable (NSC) region n

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

ID	Acce	Field	Value ID	Value	Description
A	RW	SIZE			Size of the non-secure callable (NSC) region n
			Disabled	0	The region n is not defined as a non-secure callable region. Normal security attributes (secure or non-secure) are enforced.
			32	1	The region n is defined as non-secure callable with a 32-byte size
			64	2	The region n is defined as non-secure callable with a 64-byte size
			128	3	The region n is defined as non-secure callable with a 128-byte size
			256	4	The region n is defined as non-secure callable with a 256-byte size
			512	5	The region n is defined as non-secure callable with a 512-byte size
			1024	6	The region n is defined as non-secure callable with a 1024-byte size
			2048	7	The region n is defined as non-secure callable with a 2048-byte size
			4096	8	The region n is defined as non-secure callable with a 4096-byte size
B	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset

6.17.9.20 FLASHREGION[n].PERM (n=0..31)

Address offset: 0x600 + (n × 0x4)

Access permissions for flash region n

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			E D C B A
Reset 0x00000017			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1**

ID	Acce	Field	Value ID	Value	Description
A	RW	EXECUTE			Configure instruction fetch permissions from flash region n
			Enable	1	Allow instruction fetches from flash region n
			Disable	0	Block instruction fetches from flash region n
B	RW	WRITE			Configure write permission for flash region n
			Enable	1	Allow write operation to region n
			Disable	0	Block write operation to region n
C	RW	READ			Configure read permissions for flash region n
			Enable	1	Allow read operation from flash region n
			Disable	0	Block read operation from flash region n
D	RW	SECATTR			Security attribute for flash region n
			Non_Secure	0	Flash region n security attribute is non-secure
			Secure	1	Flash region n security attribute is secure
E	RW	LOCK			
			Unlocked	0	This register can be updated

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			E D C B A
Reset 0x00000017			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1**
ID	Acce	Field	Value ID / Value / Description
			Locked / 1 / The content of this register can't be changed until the next reset

6.17.9.21 RAMREGION[n].PERM (n=0..31)

Address offset: 0x700 + (n × 0x4)

Access permissions for RAM region n

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			E D C B A
Reset 0x00000017			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1**
ID	Acce	Field	Value ID / Value / Description
A	RW	EXECUTE	Configure instruction fetch permissions from RAM region n
			Enable / 1 / Allow instruction fetches from RAM region n
			Disable / 0 / Block instruction fetches from RAM region n
B	RW	WRITE	Configure write permission for RAM region n
			Enable / 1 / Allow write operation to RAM region n
			Disable / 0 / Block write operation to RAM region n
C	RW	READ	Configure read permissions for RAM region n
			Enable / 1 / Allow read operation from RAM region n
			Disable / 0 / Block read operation from RAM region n
D	RW	SECATTR	Security attribute for RAM region n
			Non_Secure / 0 / RAM region n security attribute is non-secure
			Secure / 1 / RAM region n security attribute is secure
E	RW	LOCK	
			Unlocked / 0 / This register can be updated
			Locked / 1 / The content of this register can't be changed until the next reset

6.17.9.22 PERIPHID[n].PERM (n=0..66)

Address offset: 0x800 + (n × 0x4)

List capabilities and access permissions for the peripheral with ID n

Reset values are unique per peripheral instantation. Please refer to the peripheral instantiation table. Entries not listed in the instantiation table are undefined.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			F E D C B B A A
Reset 0x00000012			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0**
ID	Acce	Field	Value ID / Value / Description
A	R	SECUREMAPPING	Define configuration capabilities for TrustZone Cortex-M secure attribute
			Note: This field is read only
			NonSecure / 0 / This peripheral is always accessible as a non-secure peripheral
			Secure / 1 / This peripheral is always accessible as a secure peripheral

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B B A A
Reset 0x00000012				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0**
ID	Acce	Field	Value ID	Value	Description
			UserSelectable	2	Non-secure or secure attribute for this peripheral is defined by the PERIPHID[n].PERM register
			Split	3	This peripheral implements the split security mechanism. Non-secure or secure attribute for this peripheral is defined by the PERIPHID[n].PERM register.
B	R	DMA			Indicate if the peripheral has DMA capabilities and if DMA transfer can be assigned to a different security attribute than the peripheral itself
			NoDMA	0	Peripheral has no DMA capability
			NoSeparateAttribute	1	Peripheral has DMA and DMA transfers always have the same security attribute as assigned to the peripheral
			SeparateAttribute	2	Peripheral has DMA and DMA transfers can have a different security attribute than the one assigned to the peripheral
C	RW	SECATTR			Peripheral security mapping **Note:** This bit has effect only if PERIPHID[n].PERM.SECUREMAPPING reads as UserSelectable or Split
			Secure	1	Peripheral is mapped in secure peripheral address space
			NonSecure	0	If SECUREMAPPING == UserSelectable: Peripheral is mapped in non-secure peripheral address space. If SECUREMAPPING == Split: Peripheral is mapped in non-secure and secure peripheral address space.
D	RW	DMASEC			Security attribution for the DMA transfer **Note:** This bit has effect only if PERIPHID[n].PERM.SECATTR is set to secure
			Secure	1	DMA transfers initiated by this peripheral have the secure attribute set
			NonSecure	0	DMA transfers initiated by this peripheral have the non-secure attribute set
E	RW	LOCK			
			Unlocked	0	This register can be updated
			Locked	1	The content of this register can't be changed until the next reset
F	R	PRESENT			Indicate if a peripheral is present with ID n
			NotPresent	0	Peripheral is not present
			IsPresent	1	Peripheral is present

6.18 TIMER — Timer/counter

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

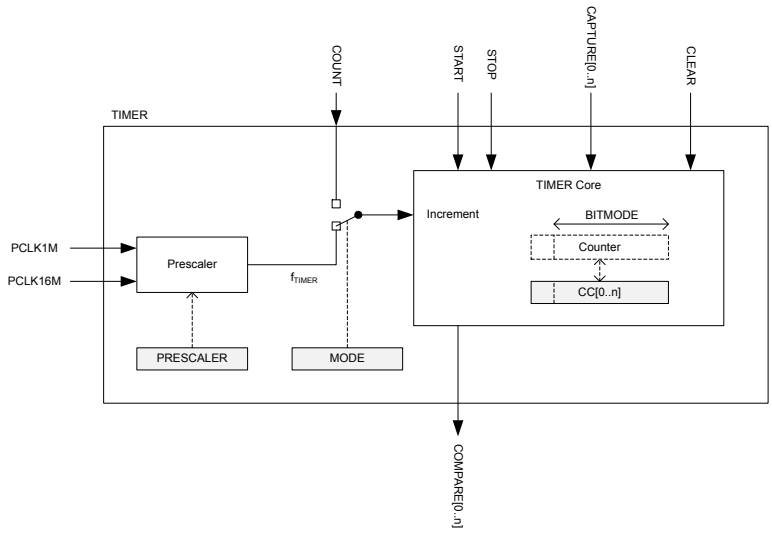

Figure 86: Block schematic for timer/counter

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

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

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

In Timer mode, the TIMER's internal Counter register is incremented by one for every tick of the timer frequency f_{TIMER} as illustrated in Block schematic for timer/counter on page 281. The timer frequency is derived from PCLK16M as shown below, using the values specified in the PRESCALER register:

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

When f_{TIMER} <= 1 MHz the TIMER will use PCLK1M instead of PCLK16M for reduced power consumption.

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

The TIMER's maximum value is configured by changing the bit-width of the timer in the BITMODE on page 288 register.

PRESCALER on page 289 and the BITMODE on page 288 must only be updated when the timer is stopped. If these registers are updated while the TIMER is started then this may result in unpredictable behavior.

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

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

The TIMER implements multiple capture/compare registers.

Independent of prescaler setting the accuracy of the TIMER is equivalent to one tick of the timer frequency f_{TIMER} as illustrated in Block schematic for timer/counter on page 281.

6.18.1 Capture

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

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

6.18.2 Compare

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

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

BITMODE on page 288 specifies how many bits of the Counter register and the capture/compare register that are used when the comparison is performed. Other bits will be ignored.

6.18.3 Task delays

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

6.18.4 Task priority

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

6.18.5 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x5000F000 0x4000F000	TIMER	TIMER0 : S TIMER0 : NS	US	NA	Timer 0	
0x50010000 0x40010000	TIMER	TIMER1 : S TIMER1 : NS	US	NA	Timer 1	
0x50011000 0x40011000	TIMER	TIMER2 : S TIMER2 : NS	US	NA	Timer 2	

Table 82: Instances

Register	Offset	Security	Description	
TASKS_START	0x000		Start Timer	
TASKS_STOP	0x004		Stop Timer	
TASKS_COUNT	0x008		Increment Timer (Counter mode only)	
TASKS_CLEAR	0x00C		Clear time	
TASKS_SHUTDOWN	0x010		Shut down timer	Deprecated
TASKS_CAPTURE[0]	0x040		Capture Timer value to CC[0] register	
TASKS_CAPTURE[1]	0x044		Capture Timer value to CC[1] register	
TASKS_CAPTURE[2]	0x048		Capture Timer value to CC[2] register	

Register	Offset	Security	Description	
TASKS_CAPTURE[3]	0x04C		Capture Timer value to CC[3] register	
TASKS_CAPTURE[4]	0x050		Capture Timer value to CC[4] register	
TASKS_CAPTURE[5]	0x054		Capture Timer value to CC[5] register	
SUBSCRIBE_START	0x080		Subscribe configuration for task START	
SUBSCRIBE_STOP	0x084		Subscribe configuration for task STOP	
SUBSCRIBE_COUNT	0x088		Subscribe configuration for task COUNT	
SUBSCRIBE_CLEAR	0x08C		Subscribe configuration for task CLEAR	
SUBSCRIBE_SHUTDOWN	0x090		Subscribe configuration for task SHUTDOWN	Deprecated
SUBSCRIBE_CAPTURE[0]	0x0C0		Subscribe configuration for task CAPTURE[0]	
SUBSCRIBE_CAPTURE[1]	0x0C4		Subscribe configuration for task CAPTURE[1]	
SUBSCRIBE_CAPTURE[2]	0x0C8		Subscribe configuration for task CAPTURE[2]	
SUBSCRIBE_CAPTURE[3]	0x0CC		Subscribe configuration for task CAPTURE[3]	
SUBSCRIBE_CAPTURE[4]	0x0D0		Subscribe configuration for task CAPTURE[4]	
SUBSCRIBE_CAPTURE[5]	0x0D4		Subscribe configuration for task CAPTURE[5]	
EVENTS_COMPARE[0]	0x140		Compare event on CC[0] match	
EVENTS_COMPARE[1]	0x144		Compare event on CC[1] match	
EVENTS_COMPARE[2]	0x148		Compare event on CC[2] match	
EVENTS_COMPARE[3]	0x14C		Compare event on CC[3] match	
EVENTS_COMPARE[4]	0x150		Compare event on CC[4] match	
EVENTS_COMPARE[5]	0x154		Compare event on CC[5] match	
PUBLISH_COMPARE[0]	0x1C0		Publish configuration for event COMPARE[0]	
PUBLISH_COMPARE[1]	0x1C4		Publish configuration for event COMPARE[1]	
PUBLISH_COMPARE[2]	0x1C8		Publish configuration for event COMPARE[2]	
PUBLISH_COMPARE[3]	0x1CC		Publish configuration for event COMPARE[3]	
PUBLISH_COMPARE[4]	0x1D0		Publish configuration for event COMPARE[4]	
PUBLISH_COMPARE[5]	0x1D4		Publish configuration for event COMPARE[5]	
SHORTS	0x200		Shortcuts between local events and tasks	
INTENSET	0x304		Enable interrupt	
INTENCLR	0x308		Disable interrupt	
MODE	0x504		Timer mode selection	
BITMODE	0x508		Configure the number of bits used by the TIMER	
PRESCALER	0x510		Timer prescaler register	
CC[0]	0x540		Capture/Compare register 0	
CC[1]	0x544		Capture/Compare register 1	
CC[2]	0x548		Capture/Compare register 2	
CC[3]	0x54C		Capture/Compare register 3	
CC[4]	0x550		Capture/Compare register 4	
CC[5]	0x554		Capture/Compare register 5	

Table 83: Register overview

6.18.5.1 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_START			Start Timer
			Trigger	1	Trigger task

6.18.5.2 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop Timer
			Trigger	1	Trigger task

6.18.5.3 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_COUNT			Increment Timer (Counter mode only)
			Trigger	1	Trigger task

6.18.5.4 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_CLEAR			Clear time
			Trigger	1	Trigger task

6.18.5.5 TASKS_SHUTDOWN (Deprecated)

Address offset: 0x010

Shut down timer

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

ID	Acce	Field	Value ID	Value	Description	
A	W	TASKS_SHUTDOWN			Shut down timer	Deprecated
			Trigger	1	Trigger task	

6.18.5.6 TASKS_CAPTURE[n] (n=0..5)

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	Acce	Field	Value ID	Value	Description
A	W	TASKS_CAPTURE			Capture Timer value to CC[n] register
			Trigger	1	Trigger task

6.18.5.7 SUBSCRIBE_START

Address offset: 0x080

Subscribe configuration for task START

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.18.5.8 SUBSCRIBE_STOP

Address offset: 0x084

Subscribe configuration for task STOP

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.18.5.9 SUBSCRIBE_COUNT

Address offset: 0x088

Subscribe configuration for task COUNT

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task COUNT will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.18.5.10 SUBSCRIBE_CLEAR

Address offset: 0x08C

Subscribe configuration for task CLEAR

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

6.18.5.11 SUBSCRIBE_SHUTDOWN (Deprecated)

Address offset: 0x090

Subscribe configuration for task SHUTDOWN

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

6.18.5.12 SUBSCRIBE_CAPTURE[n] (n=0..5)

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task CAPTURE[n] will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.18.5.13 EVENTS_COMPARE[n] (n=0..5)

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	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_COMPARE			Compare event on CC[n] match
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.18.5.14 PUBLISH_COMPARE[n] (n=0..5)

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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event COMPARE[n] will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.18.5.15 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				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	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i]_CLEAR (i=0..5)			Shortcut between event COMPARE[i] and task CLEAR
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
G-L	RW	COMPARE[i]_STOP (i=0..5)			Shortcut between event COMPARE[i] and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.18.5.16 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i] (i=0..5)			Write '1' to enable interrupt for event COMPARE[i]
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.5.17 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A-F	RW	COMPARE[i] (i=0..5)			Write '1' to disable interrupt for event COMPARE[i]
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.18.5.18 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	Acce	Field	Value ID	Value	Description
A	RW	MODE			Timer mode
			Timer	0	Select Timer mode
			Counter	1	Select Counter mode
			LowPowerCounter	2	Select Low Power Counter mode

6.18.5.19 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	Acce	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

6.18.5.20 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	Acce	Field	Value ID	Value	Description
A	RW	PRESCALER		[0..9]	Prescaler value

6.18.5.21 CC[n] (n=0..5)

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	Acce	Field	Value ID	Value	Description
A	RW	CC			Capture/Compare value Only the number of bits indicated by BITMODE will be used by the TIMER.

6.18.6 Electrical specification

6.19 TWIM — I^2C compatible two-wire interface master with EasyDMA

TWI master with EasyDMA (TWIM) is a two-wire half-duplex master which can communicate with multiple slave devices connected to the same bus

Listed here are the main features for TWIM:

- I^2C compatible
- 100 kbps, 250 kbps, or 400 kbps
- Support for clock stretching (non I^2C compliant)
- EasyDMA

The two-wire interface can communicate with a bi-directional wired-AND bus with two lines (SCL, SDA). The protocol makes it possible to interconnect up to 127 individually addressable devices. TWIM is not compatible with CBUS.

The GPIOs used for each two-wire interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

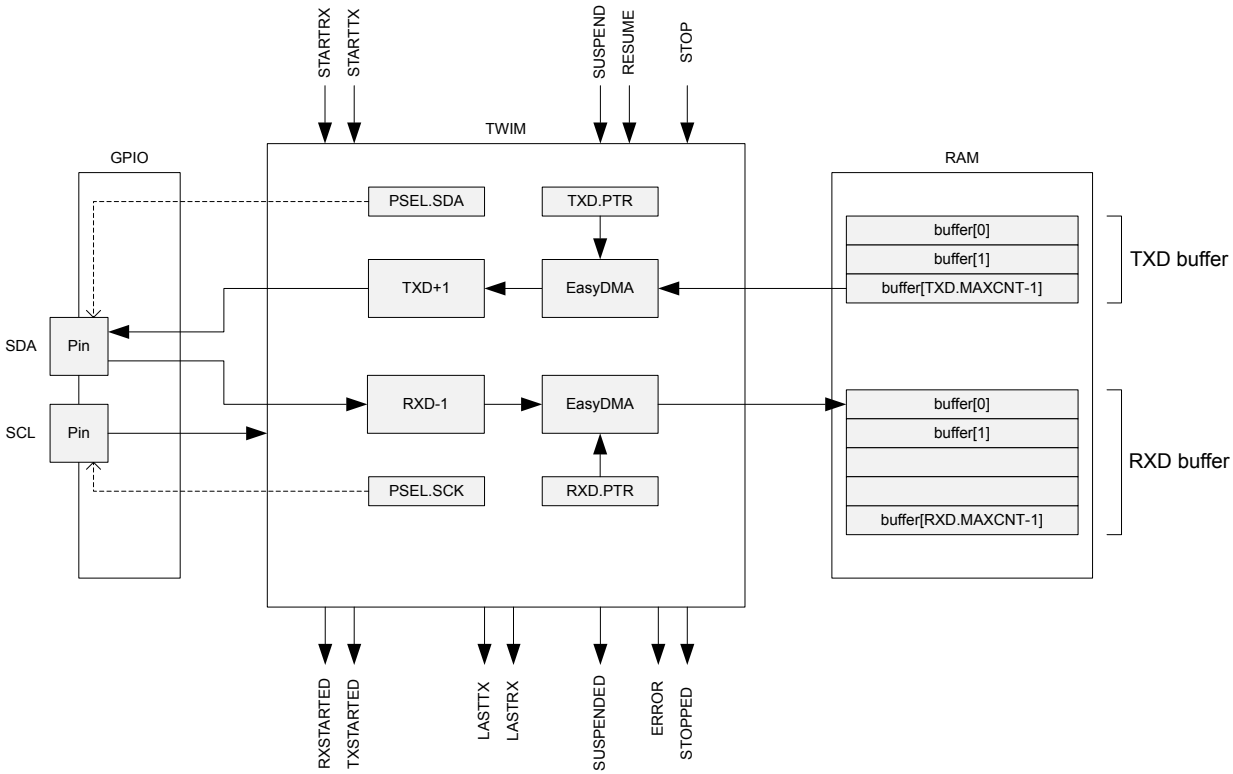

Figure 87: TWI master with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 290. This TWIM is only able to operate as a single master on the TWI bus. Multi-master bus configuration is not supported.

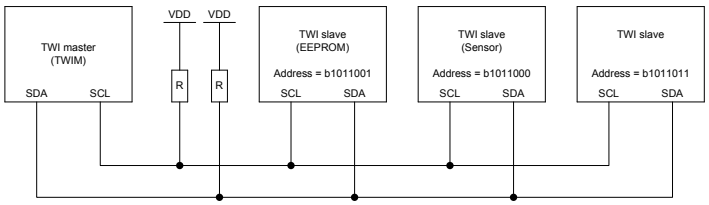

Figure 88: A typical TWI setup comprising one master and three slaves

This TWI master supports clock stretching performed by the slaves. Note that the SCK pulse following a stretched clock cycle may be shorter than specified by the I2C specification.

The TWI master is started by triggering the STARTTX or STARTRX tasks, and stopped by triggering the STOP task. The TWI master will generate a STOPPED event when it has stopped following a STOP task. The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

After the TWI master is started, the STARTTX task or the STARTRX task should not be triggered again before the TWI master has stopped, i.e. following a LASTRX, LASTTX or STOPPED event.

If a NACK is clocked in from the slave, the TWI master will generate an ERROR event.

6.19.1 EasyDMA

The TWI master implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

6.19.2 Master write sequence

A TWI master write sequence is started by triggering the STARTTX task. After the STARTTX task has been triggered, the TWI master will generate a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1).

The address must match the address of the slave device that the master wants to write to. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) generated by the slave.

After receiving the ACK bit, the TWI master will clock out the data bytes found in the transmit buffer located in RAM at the address specified in the TXD.PTR register. Each byte clocked out from the master will be followed by an ACK/NACK bit clocked in from the slave.

A typical TWI master write sequence is illustrated in TWI master writing data to a slave on page 291. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

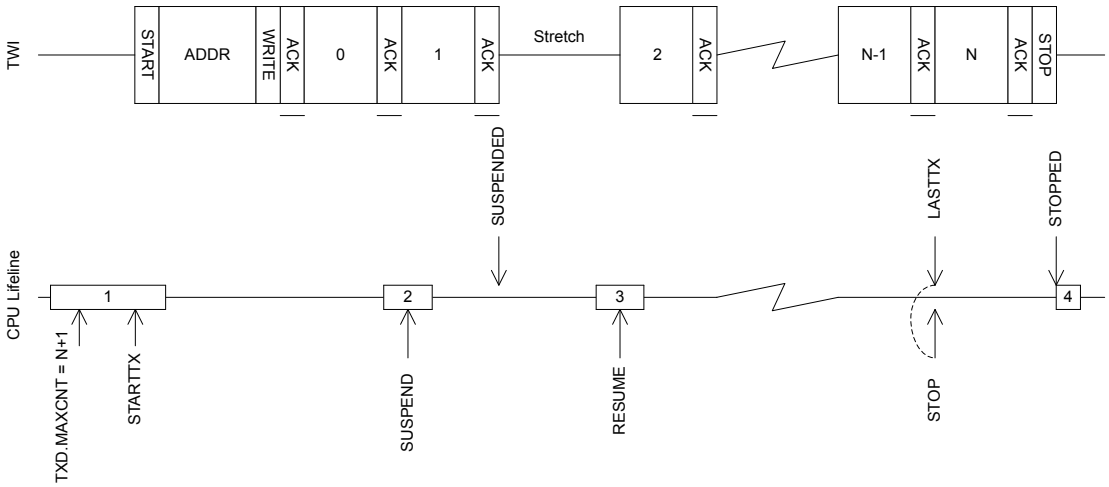

Figure 89: TWI master writing data to a slave

The TWI master will generate a LASTTX event when it starts to transmit the last byte, this is illustrated in TWI master writing data to a slave on page 291

The TWI master is stopped by triggering the STOP task, this task should be triggered during the transmission of the last byte to secure that the TWI will stop as fast as possible after sending the last byte. It is safe to use the shortcut between LASTTX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the whole RAM buffer has been sent, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

6.19.3 Master read sequence

A TWI master read sequence is started by triggering the STARTRX task. After the STARTRX task has been triggered the TWI master will generate a start condition on the TWI bus, followed by clocking out the

address and the READ/WRITE bit set to 1 (WRITE = 0, READ = 1). The address must match the address of the slave device that the master wants to read from. The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK = 1) generated by the slave.

After having sent the ACK bit the TWI slave will send data to the master using the clock generated by the master.

Data received will be stored in RAM at the address specified in the RXD.PTR register. The TWI master will generate an ACK after all but the last byte received from the slave. The TWI master will generate a NACK after the last byte received to indicate that the read sequence shall stop.

A typical TWI master read sequence is illustrated in The TWI master reading data from a slave on page 292. Occurrence 2 in the figure illustrates clock stretching performed by the TWI master following a SUSPEND task.

A SUSPENDED event indicates that the SUSPEND task has taken effect; this event can be used to synchronize the software.

The TWI master will generate a LASTRX event when it is ready to receive the last byte, this is illustrated in The TWI master reading data from a slave on page 292. If RXD.MAXCNT > 1 the LASTRX event is generated after sending the ACK of the previously received byte. If RXD.MAXCNT = 1 the LASTRX event is generated after receiving the ACK following the address and READ bit.

The TWI master is stopped by triggering the STOP task, this task must be triggered before the NACK bit is supposed to be transmitted. The STOP task can be triggered at any time during the reception of the last byte. It is safe to use the shortcut between LASTRX and STOP to accomplish this.

Note that the TWI master does not stop by itself when the RAM buffer is full, or when an error occurs. The STOP task must be issued, through the use of a local or PPI shortcut, or in software as part of the error handler.

The TWI master cannot get stopped while it is suspended, so the STOP task has to be issued after the TWI master has been resumed.

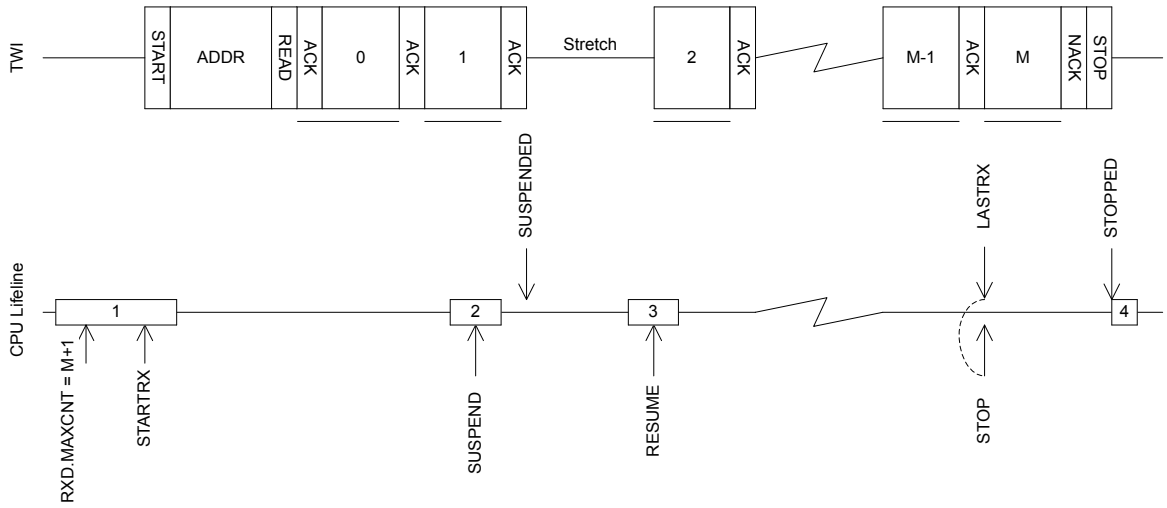

Figure 90: The TWI master reading data from a slave

6.19.4 Master repeated start sequence

A typical repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave. This example uses shortcuts to perform the simplest type of repeated start sequence, i.e. one write followed by one read. The same approach can be used to perform a repeated start sequence where the sequence is read followed by write.

The figure A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave on page 293 illustrates this:

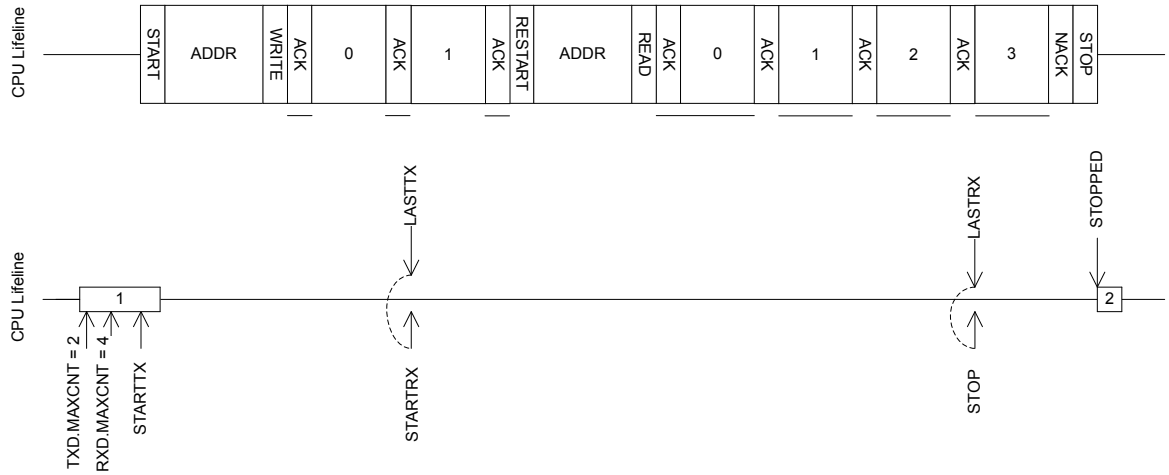

Figure 91: A repeated start sequence, where the TWI master writes two bytes followed by reading 4 bytes from the slave

If a more complex repeated start sequence is needed and the TWI firmware drive is serviced in a low priority interrupt it may be necessary to use the SUSPEND task and SUSPENDED event to guarantee that the correct tasks are generated at the correct time. This is illustrated in A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts on page 293.

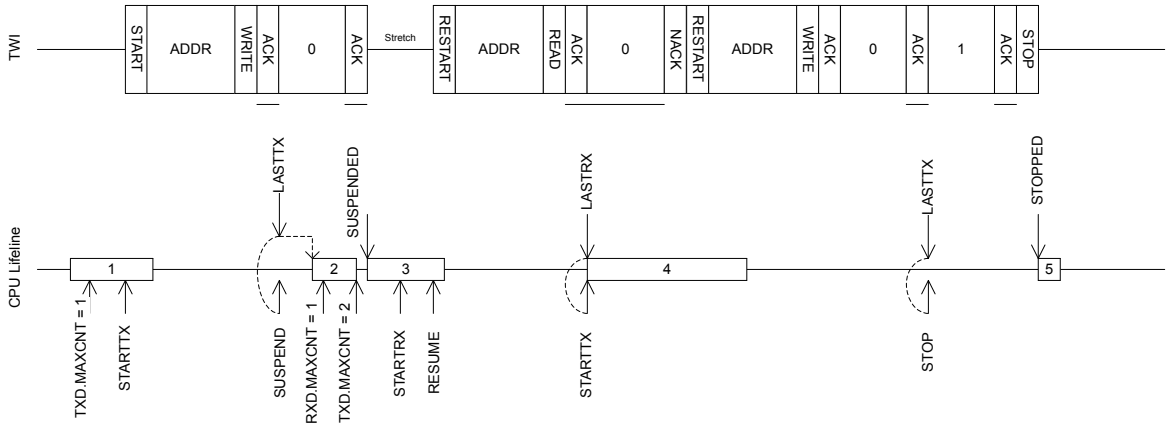

Figure 92: A double repeated start sequence using the SUSPEND task to secure safe operation in low priority interrupts

6.19.5 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.19.6 Master mode pin configuration

The SCL and SDA signals associated with the TWI master are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI master is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins

will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL, PSEL.SDA must only be configured when the TWI master is disabled.

To secure correct signal levels on the pins used by the TWI master when the system is in OFF mode, and when the TWI master is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 294.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI master signal	TWI master pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 84: GPIO configuration before enabling peripheral

6.19.7 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	TWIM	TWIM0 : S TWIM0 : NS	US	SA	Two-wire interface master 0	
0x50009000 0x40009000	TWIM	TWIM1 : S TWIM1 : NS	US	SA	Two-wire interface master 1	
0x5000A000 0x4000A000	TWIM	TWIM2 : S TWIM2 : NS	US	SA	Two-wire interface master 2	
0x5000B000 0x4000B000	TWIM	TWIM3 : S TWIM3 : NS	US	SA	Two-wire interface master 3	

Table 85: Instances

Register	Offset	Security	Description
TASKS_STARTRX	0x000		Start TWI receive sequence
TASKS_STARTTX	0x008		Start TWI transmit sequence
TASKS_STOP	0x014		Stop TWI transaction. Must be issued while the TWI master is not suspended.
TASKS_SUSPEND	0x01C		Suspend TWI transaction
TASKS_RESUME	0x020		Resume TWI transaction
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
EVENTS_STOPPED	0x104		TWI stopped
EVENTS_ERROR	0x124		TWI error
EVENTS_SUSPENDED	0x148		Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
EVENTS_RXSTARTED	0x14C		Receive sequence started
EVENTS_TXSTARTED	0x150		Transmit sequence started
EVENTS_LASTRX	0x15C		Byte boundary, starting to receive the last byte
EVENTS_LASTTX	0x160		Byte boundary, starting to transmit the last byte
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_SUSPENDED	0x1C8		Publish configuration for event SUSPENDED
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED

Register	Offset	Security	Description
PUBLISH_LASTRX	0x1DC		Publish configuration for event LASTRX
PUBLISH_LASTTX	0x1E0		Publish configuration for event LASTTX
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4C4		Error source
ENABLE	0x500		Enable TWIM
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal
FREQUENCY	0x524		TWI frequency. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
RXD.LIST	0x540		EasyDMA list type
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
TXD.LIST	0x550		EasyDMA list type
ADDRESS	0x588		Address used in the TWI transfer

Table 86: Register overview

6.19.7.1 TASKS_STARTRX

Address offset: 0x000

Start TWI receive 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTRX			Start TWI receive sequence
			Trigger	1	Trigger task

6.19.7.2 TASKS_STARTTX

Address offset: 0x008

Start TWI transmit 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	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start TWI transmit sequence
			Trigger	1	Trigger task

6.19.7.3 TASKS_STOP

Address offset: 0x014

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	Acce	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

6.19.7.4 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

6.19.7.5 TASKS_RESUME

Address offset: 0x020

Resume TWI transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

6.19.7.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

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

6.19.7.7 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STARTTX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.8 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task STOP will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.9 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task SUSPEND will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.10 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.19.7.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.13 EVENTS_SUSPENDED

Address offset: 0x148

Last byte has been sent out after the 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	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_SUSPENDED			Last byte has been sent out after the SUSPEND task has been issued, TWI traffic is now suspended.
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.14 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.15 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.19.7.16 EVENTS_LASTRX

Address offset: 0x15C

Byte boundary, starting to receive the last byte

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

6.19.7.17 EVENTS_LASTTX

Address offset: 0x160

Byte boundary, starting to transmit the last byte

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

6.19.7.18 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
Reset 0x00000000			**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event STOPPED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.19 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.20 PUBLISH_SUSPENDED

Address offset: 0x1C8

Publish configuration for event SUSPENDED

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event SUSPENDED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.21 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.19.7.22 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

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

6.19.7.23 PUBLISH_LASTRX

Address offset: 0x1DC

Publish configuration for event LASTRX

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

6.19.7.24 PUBLISH_LASTTX

Address offset: 0x1E0

Publish configuration for event LASTTX

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

6.19.7.25 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LASTTX_STARTRX			Shortcut between event LASTTX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
B	RW	LASTTX_SUSPEND			Shortcut between event LASTTX and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
C	RW	LASTTX_STOP			Shortcut between event LASTTX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	LASTRX_STARTTX			Shortcut between event LASTRX and task STARTTX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
E	RW	LASTRX_SUSPEND			Shortcut between event LASTRX and task SUSPEND
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
F	RW	LASTRX_STOP			Shortcut between event LASTRX and task STOP
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.19.7.26 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	SUSPENDED			Enable or disable interrupt for event SUSPENDED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	LASTRX			Enable or disable interrupt for event LASTRX
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	LASTTX			Enable or disable interrupt for event LASTTX
			Disabled	0	Disable
			Enabled	1	Enable

6.19.7.27 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to enable interrupt for event SUSPENDED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to enable interrupt for event LASTRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to enable interrupt for event LASTTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.7.28 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				J I H G F D A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	SUSPENDED			Write '1' to disable interrupt for event SUSPENDED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	LASTRX			Write '1' to disable interrupt for event LASTRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	LASTTX			Write '1' to disable interrupt for event LASTTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.19.7.29 ERRORSRC

Address offset: 0x4C4

Error source

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A new byte was received before previous byte got transferred into RXD buffer. (Previous data is lost)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
B	RW	ANACK			NACK received after sending the address (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	DNACK			NACK received after sending a data byte (write '1' to clear)
			NotReceived	0	Error did not occur
			Received	1	Error occurred

6.19.7.30 ENABLE

Address offset: 0x500

Enable TWIM

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIM
			Disabled	0	Disable TWIM
			Enabled	6	Enable TWIM

6.19.7.31 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.7.32 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.19.7.33 FREQUENCY

Address offset: 0x524

TWI frequency. Accuracy depends on the HFCLK source selected.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000				**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	FREQUENCY			TWI master clock frequency
			K100	0x01980000	100 kbps
			K250	0x04000000	250 kbps
			K400	0x06400000	400 kbps

6.19.7.34 RXD.PTR

Address offset: 0x534

Data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.19.7.35 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.19.7.36 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.19.7.37 RXD.LIST

Address offset: 0x540

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.19.7.38 TXD.PTR

Address offset: 0x544

Data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.19.7.39 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.19.7.40 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A A A A A A A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction. In case of NACK error, includes the NACK'ed byte.

6.19.7.41 TXD.LIST

Address offset: 0x550

EasyDMA list type

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	LIST			List type
			Disabled	0	Disable EasyDMA list
			ArrayList	1	Use array list

6.19.7.42 ADDRESS

Address offset: 0x588

Address used in the TWI transfer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	ADDRESS			Address used in the TWI transfer

6.19.8 Electrical specification

6.19.8.1 TWIM interface electrical specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIM,SCL}$	Bit rates for TWIM[14]	100		400	kbps
$t_{TWIM,START}$	Time from STARTRX/STARTTX task to transmission started	..	..	..	µs

6.19.8.2 Two Wire Interface Master (TWIM) timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
t_{TWIM,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIM,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			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, 250kbps	4000			ns
$t_{TWIM,HD\_STA,400kbps}$	TWIM master hold time for START and repeated START condition, 400 kbps	2500			ns
$t_{TWIM,SU\_STO,100kbps}$	TWIM master setup time from SCL high to STOP condition, 100 kbps	5000			ns
$t_{TWIM,SU\_STO,250kbps}$	TWIM master setup time from SCL high to STOP condition, 250 kbps	2000			ns
$t_{TWIM,SU\_STO,400kbps}$	TWIM master setup time from SCL high to STOP condition, 400 kbps	1250			ns
$t_{TWIM,BUF,100kbps}$	TWIM master bus free time between STOP and START conditions, 100 kbps	5800			ns
$t_{TWIM,BUF,250kbps}$	TWIM master bus free time between STOP and START conditions, 250 kbps	2700			ns
$t_{TWIM,BUF,400kbps}$	TWIM master bus free time between STOP and START conditions, 400 kbps	2100			ns

[14] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

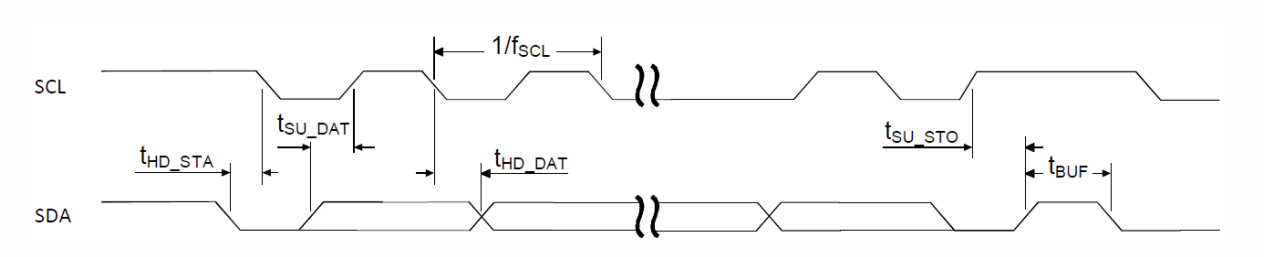

Figure 93: TWIM timing diagram, 1 byte transaction

6.19.9 Pullup resistor

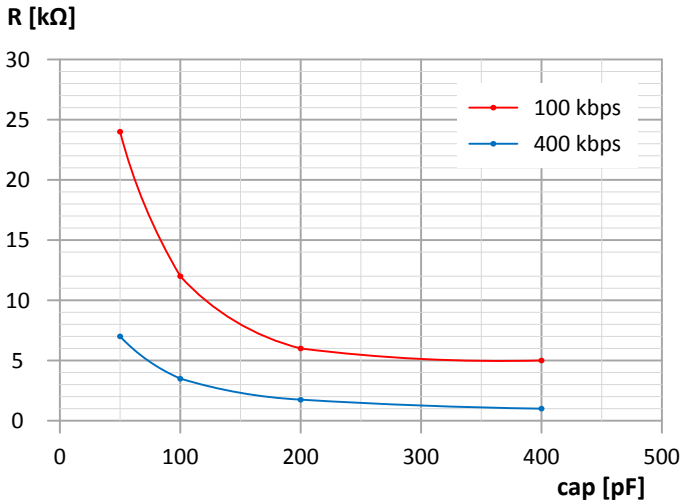

Figure 94: Recommended TWIM pullup value vs. line capacitance

- The I2C specification allows a line capacitance of 400 pF at most.
- The value of internal pullup resistor (R_{PU}) for nRF9160 can be found in GPIO — General purpose input/output on page 91.

6.20 TWIS — I²C compatible two-wire interface slave with EasyDMA

TWI slave with EasyDMA (TWIS) is compatible with I²C operating at 100 kHz and 400 kHz. The TWI transmitter and receiver implement EasyDMA.

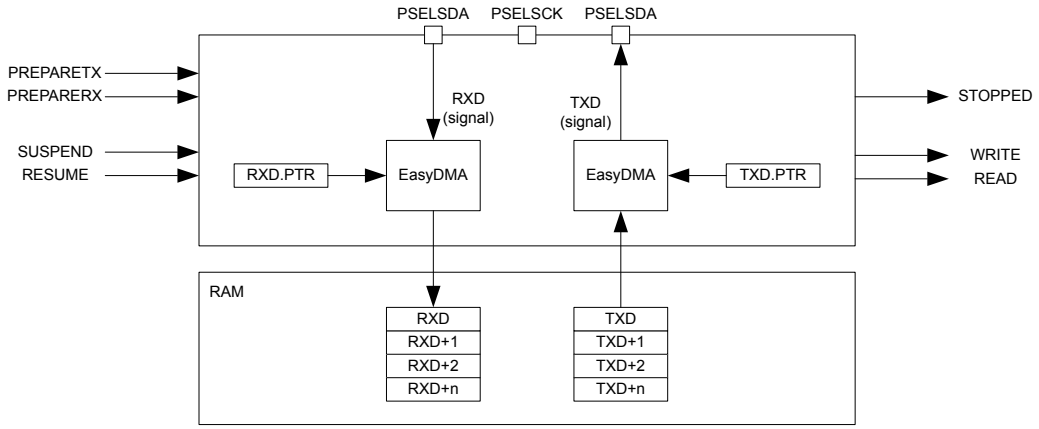

Figure 95: TWI slave with EasyDMA

A typical TWI setup consists of one master and one or more slaves. For an example, see A typical TWI setup comprising one master and three slaves on page 310. TWIS is only able to operate with a single master on the TWI bus.

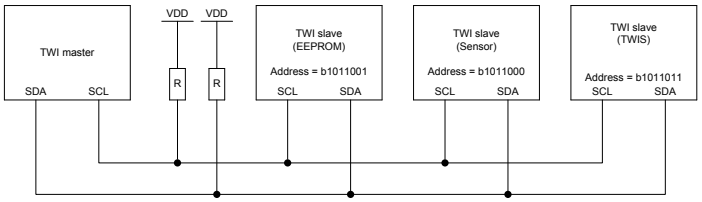

Figure 96: A typical TWI setup comprising one master and three slaves

The TWI slave state machine is illustrated in TWI slave state machine on page 311 and TWI slave state machine symbols on page 311 is explaining the different symbols used in the state machine.

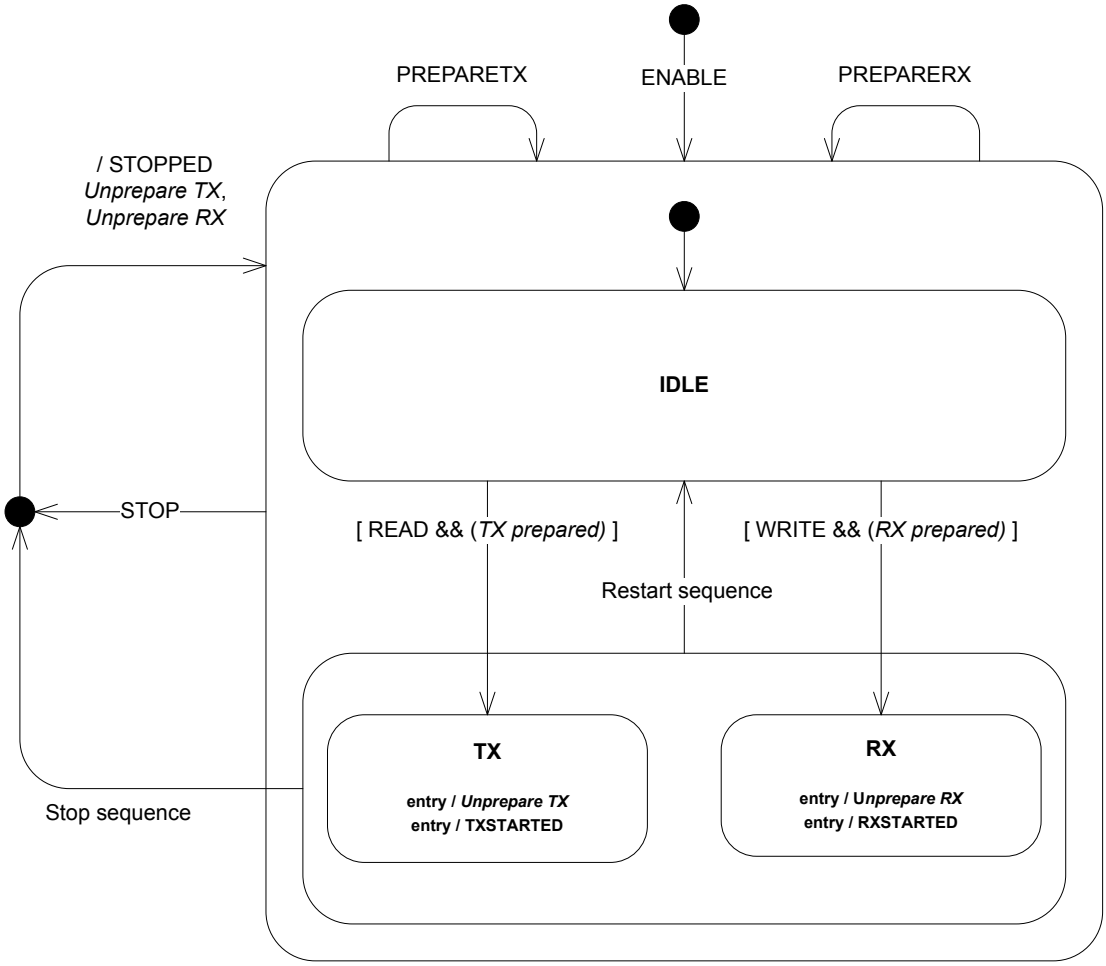

Figure 97: TWI slave state machine

Symbol	Type	Description
ENABLE	Register	The TWI slave has been enabled via the ENABLE register
PREPARETX	Task	The TASKS_PREPARETX task has been triggered
STOP	Task	The TASKS_STOP task has been triggered
PREPARERX	Task	The TASKS_PREPARERX task has been triggered
STOPPED	Event	The EVENTS_STOPPED event was generated
RXSTARTED	Event	The EVENTS_RXSTARTED event was generated
TXSTARTED	Event	The EVENTS_TXSTARTED event was generated
TX prepared	Internal	Internal flag indicating that a TASKS_PREPARETX task has been triggered. This flag is not visible to the user.
RX prepared	Internal	Internal flag indicating that a TASKS_PREPARERX task has been triggered. This flag is not visible to the user.
Unprepare TX	Internal	Clears the internal 'TX prepared' flag until next TASKS_PREPARETX task.
Unprepare RX	Internal	Clears the internal 'RX prepared' flag until next TASKS_PREPARERX task.
Stop sequence	TWI protocol	A TWI stop sequence was detected
Restart sequence	TWI protocol	A TWI restart sequence was detected

Table 87: TWI slave state machine symbols

The TWI slave supports clock stretching performed by the master.

The TWI slave operates in a low power mode while waiting for a TWI master to initiate a transfer. As long as the TWI slave is not addressed, it will remain in this low power mode.

To secure correct behaviour of the TWI slave, PSEL.SCL, PSEL.SDA, CONFIG and the ADDRESS[n] registers, must be configured prior to enabling the TWI slave through the ENABLE register. Similarly, changing these settings must be performed while the TWI slave is disabled. Failing to do so may result in unpredictable behaviour.

6.20.1 EasyDMA

The TWI slave implements EasyDMA for reading and writing to and from the RAM.

The STOPPED event indicates that EasyDMA has finished accessing the buffer in RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

6.20.2 TWI slave responding to a read command

Before the TWI slave can respond to a read command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE} .

A read command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 1 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the TWI slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the read command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a READ event when it acknowledges the read command.

The TWI slave is only able to detect a read command from the IDLE state.

The TWI slave will set an internal 'TX prepared' flag when the PREPARETX task is triggered.

When the read command is received the TWI slave will enter the TX state if the internal 'TX prepared' flag is set.

If the internal 'TX prepared' flag is not set when the read command is received, the TWI slave will stretch the master's clock until the PREPARETX task is triggered and the internal 'TX prepared' flag is set.

The TWI slave will generate the TXSTARTED event and clear the 'TX prepared' flag ('unprepare TX') when it enters the TX state. In this state the TWI slave will send the data bytes found in the transmit buffer to the master using the master's clock. The TWI slave will consume I_{TX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the TX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the 'TX prepared' flag ('unprepare TX') and go back to the IDLE state when it has stopped.

The transmit buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to send TXD.MAXCNT bytes from the transmit buffer for each transaction. If the TWI master forces the slave to send more than TXD.MAXCNT bytes, the slave will send the byte specified in the ORC register to the master instead. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see TXD.PTR etc., are latched when the TXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the 'TX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 315.

Each byte sent from the slave will be followed by an ACK/NACK bit sent from the master. The TWI master will generate a NACK following the last byte that it wants to receive to tell the slave to release the bus so that the TWI master can generate the stop condition. The TXD.AMOUNT register can be queried after a transaction to see how many bytes were sent.

A typical TWI slave read command response is illustrated in The TWI slave responding to a read command on page 313. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

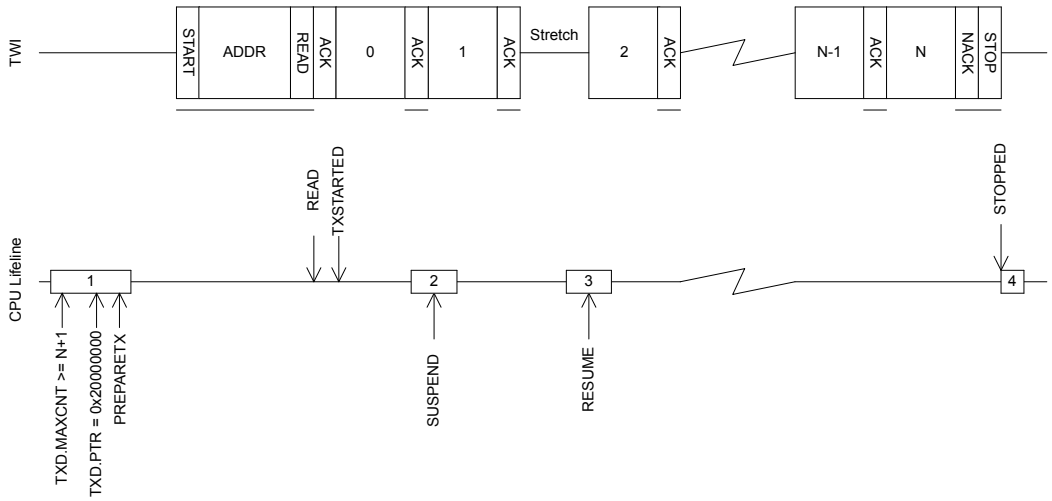

Figure 98: The TWI slave responding to a read command

6.20.3 TWI slave responding to a write command

Before the TWI slave can respond to a write command the TWI slave must be configured correctly and enabled via the ENABLE register. When enabled the TWI slave will be in its IDLE state where it will consume I_{IDLE}.

A write command is started when the TWI master generates a start condition on the TWI bus, followed by clocking out the address and the READ/WRITE bit set to 0 (WRITE=0, READ=1). The READ/WRITE bit is followed by an ACK/NACK bit (ACK=0 or NACK=1) response from the slave.

The TWI slave is able to listen for up to two addresses at the same time. Which addresses to listen for is configured in the ADDRESS registers and the CONFIG register.

The TWI slave will only acknowledge (ACK) the write command if the address presented by the master matches one of the addresses the slave is configured to listen for. The TWI slave will generate a WRITE event if it acknowledges the write command.

The TWI slave is only able to detect a write command from the IDLE state.

The TWI slave will set an internal 'RX prepared' flag when the PREPARERX task is triggered.

When the write command is received the TWI slave will enter the RX state if the internal 'RX prepared' flag is set.

If the internal 'RX prepared' flag is not set when the write command is received, the TWI slave will stretch the master's clock until the PREPARERX task is triggered and the internal 'RX prepared' flag is set.

The TWI slave will generate the RXSTARTED event and clear the internal 'RX prepared' flag ('unprepare RX') when it enters the RX state. In this state the TWI slave will be able to receive the bytes sent by the TWI master. The TWI slave will consume I_{RX} in this mode.

The TWI slave will go back to the IDLE state if the TWI slave receives a restart command when it is in the RX state.

The TWI slave is stopped when it receives the stop condition from the TWI master. A STOPPED event will be generated when the transaction has stopped. The TWI slave will clear the internal 'RX prepared' flag ('unprepare RX') and go back to the IDLE state when it has stopped.

The receive buffer is located in RAM at the address specified in the TXD.PTR register. The TWI slave will only be able to receive as many bytes as specified in the RXD.MAXCNT register. If the TWI master tries to send more bytes to the slave than the slave is able to receive,these bytes will be discarded and the bytes will be NACKed by the slave. If this happens, an ERROR event will be generated.

The EasyDMA configuration registers, see RXD.PTR etc., are latched when the RXSTARTED event is generated.

The TWI slave can be forced to stop by triggering the STOP task. A STOPPED event will be generated when the TWI slave has stopped. The TWI slave will clear the internal 'RX prepared' flag and go back to the IDLE state when it has stopped, see also Terminating an ongoing TWI transaction on page 315.

The TWI slave will generate an ACK after every byte received from the master. The RXD.AMOUNT register can be queried after a transaction to see how many bytes were received.

A typical TWI slave write command response is illustrated in The TWI slave responding to a write command on page 314. Occurrence 2 in the figure illustrates clock stretching performed by the TWI slave following a SUSPEND task.

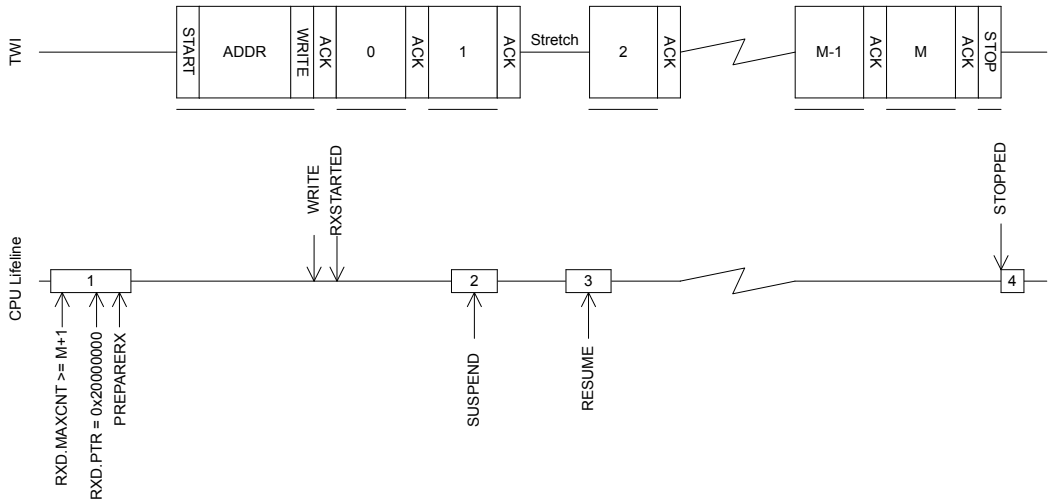

Figure 99: The TWI slave responding to a write command

6.20.4 Master repeated start sequence

An example of a repeated start sequence is one in which the TWI master writes two bytes to the slave followed by reading four bytes from the slave.

This is illustrated in A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave on page 315.

It is here assumed that the receiver does not know in advance what the master wants to read, and that this information is provided in the first two bytes received in the write part of the repeated start sequence. To guarantee that the CPU is able to process the received data before the TWI slave starts to reply to the read command, the SUSPEND task is triggered via a shortcut from the READ event generated when the read command is received. When the CPU has processed the incoming data and prepared the correct data response, the CPU will resume the transaction by triggering the RESUME task.

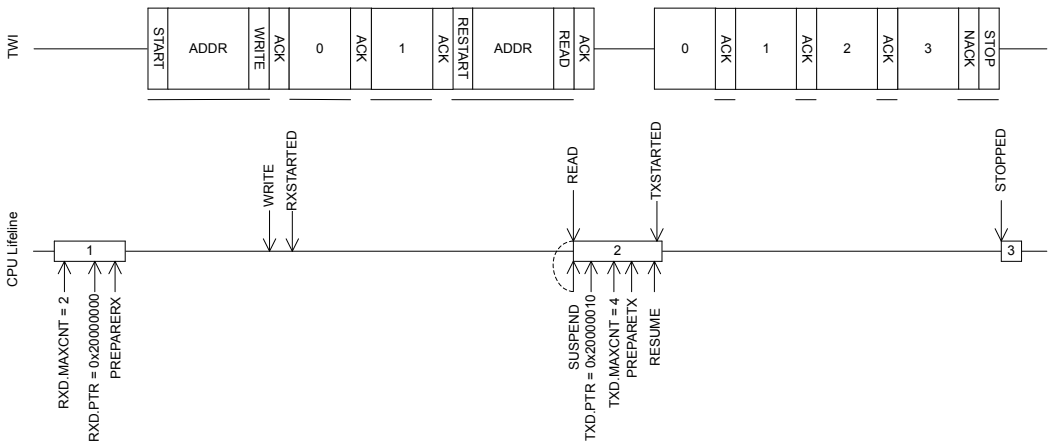

Figure 100: A repeated start sequence, where the TWI master writes two bytes followed by reading four bytes from the slave

6.20.5 Terminating an ongoing TWI transaction

In some situations, e.g. if the external TWI master is not responding correctly, it may be required to terminate an ongoing transaction.

This can be achieved by triggering the STOP task. In this situation a STOPPED event will be generated when the TWI has stopped independent of whether or not a STOP condition has been generated on the TWI bus. The TWI slave will release the bus when it has stopped and go back to its IDLE state.

6.20.6 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOP task may not be always needed (the peripheral might already be stopped), but if it is sent, software shall wait until the STOPPED event was received as a response before disabling the peripheral through the ENABLE register.

6.20.7 Slave mode pin configuration

The SCL and SDA signals associated with the TWI slave are mapped to physical pins according to the configuration specified in the PSEL.SCL and PSEL.SDA registers respectively.

The PSEL.SCL and PSEL.SDA registers and their configurations are only used as long as the TWI slave is enabled, and retained only as long as the device is in ON mode. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register. PSEL.SCL and PSEL.SDA must only be configured when the TWI slave is disabled.

To secure correct signal levels on the pins used by the TWI slave when the system is in OFF mode, and when the TWI slave is disabled, these pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 315.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

TWI slave signal	TWI slave pin	Direction	Output value	Drive strength
SCL	As specified in PSEL.SCL	Input	Not applicable	S0D1
SDA	As specified in PSEL.SDA	Input	Not applicable	S0D1

Table 88: GPIO configuration before enabling peripheral

6.20.8 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	TWIS	TWIS0 : S TWIS0 : NS	US	SA	Two-wire interface slave 0	
0x50009000 0x40009000	TWIS	TWIS1 : S TWIS1 : NS	US	SA	Two-wire interface slave 1	
0x5000A000 0x4000A000	TWIS	TWIS2 : S TWIS2 : NS	US	SA	Two-wire interface slave 2	
0x5000B000 0x4000B000	TWIS	TWIS3 : S TWIS3 : NS	US	SA	Two-wire interface slave 3	

Table 89: Instances

Register	Offset	Security	Description
TASKS_STOP	0x014		Stop TWI transaction
TASKS_SUSPEND	0x01C		Suspend TWI transaction
TASKS_RESUME	0x020		Resume TWI transaction
TASKS_PREPARERX	0x030		Prepare the TWI slave to respond to a write command
TASKS_PREPARETX	0x034		Prepare the TWI slave to respond to a read command
SUBSCRIBE_STOP	0x094		Subscribe configuration for task STOP
SUBSCRIBE_SUSPEND	0x09C		Subscribe configuration for task SUSPEND
SUBSCRIBE_RESUME	0x0A0		Subscribe configuration for task RESUME
SUBSCRIBE_PREPARERX	0x0B0		Subscribe configuration for task PREPARERX
SUBSCRIBE_PREPARETX	0x0B4		Subscribe configuration for task PREPARETX
EVENTS_STOPPED	0x104		TWI stopped
EVENTS_ERROR	0x124		TWI error
EVENTS_RXSTARTED	0x14C		Receive sequence started
EVENTS_TXSTARTED	0x150		Transmit sequence started
EVENTS_WRITE	0x164		Write command received
EVENTS_READ	0x168		Read command received
PUBLISH_STOPPED	0x184		Publish configuration for event STOPPED
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_WRITE	0x1E4		Publish configuration for event WRITE
PUBLISH_READ	0x1E8		Publish configuration for event READ
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x4D0		Error source
MATCH	0x4D4		Status register indicating which address had a match
ENABLE	0x500		Enable TWIS
PSEL.SCL	0x508		Pin select for SCL signal
PSEL.SDA	0x50C		Pin select for SDA signal
RXD.PTR	0x534		RXD Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in RXD buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last RXD transaction
TXD.PTR	0x544		TXD Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in TXD buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last TXD transaction
ADDRESS[0]	0x588		TWI slave address 0

Register	Offset	Security	Description
ADDRESS[1]	0x58C		TWI slave address 1
CONFIG	0x594		Configuration register for the address match mechanism
ORC	0x5C0		Over-read character. Character sent out in case of an over-read of the transmit buffer.

Table 90: Register overview

6.20.8.1 TASKS_STOP

Address offset: 0x014

Stop TWI transaction

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STOP			Stop TWI transaction
			Trigger	1	Trigger task

6.20.8.2 TASKS_SUSPEND

Address offset: 0x01C

Suspend TWI transaction

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_SUSPEND			Suspend TWI transaction
			Trigger	1	Trigger task

6.20.8.3 TASKS_RESUME

Address offset: 0x020

Resume TWI transaction

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

ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_RESUME			Resume TWI transaction
			Trigger	1	Trigger task

6.20.8.4 TASKS_PREPARERX

Address offset: 0x030

Prepare the TWI slave to respond to a write command

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_PREPARERX			Prepare the TWI slave to respond to a write command
			Trigger	1	Trigger task

6.20.8.5 TASKS_PREPARETX

Address offset: 0x034

Prepare the TWI slave to respond to a read command

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_PREPARETX			Prepare the TWI slave to respond to a read command
			Trigger	1	Trigger task

6.20.8.6 SUBSCRIBE_STOP

Address offset: 0x094

Subscribe configuration for task STOP

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

6.20.8.7 SUBSCRIBE_SUSPEND

Address offset: 0x09C

Subscribe configuration for task SUSPEND

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

6.20.8.8 SUBSCRIBE_RESUME

Address offset: 0x0A0

Subscribe configuration for task RESUME

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task RESUME will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.9 SUBSCRIBE_PREPARERX

Address offset: 0x0B0

Subscribe configuration for task PREPARERX

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task PREPARERX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.10 SUBSCRIBE_PREPARETX

Address offset: 0x0B4

Subscribe configuration for task PREPARETX

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task PREPARETX will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.20.8.11 EVENTS_STOPPED

Address offset: 0x104

TWI stopped

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

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_STOPPED			TWI stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.12 EVENTS_ERROR

Address offset: 0x124

TWI error

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			TWI error
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.13 EVENTS_RXSTARTED

Address offset: 0x14C

Receive sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			Receive sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.14 EVENTS_TXSTARTED

Address offset: 0x150

Transmit sequence started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			Transmit sequence started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.15 EVENTS_WRITE

Address offset: 0x164

Write command received

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_WRITE			Write command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.16 EVENTS_READ

Address offset: 0x168

Read command received

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_READ			Read command received
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.20.8.17 PUBLISH_STOPPED

Address offset: 0x184

Publish configuration for event STOPPED

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

6.20.8.18 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

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

6.20.8.19 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

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

6.20.8.20 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

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

6.20.8.21 PUBLISH_WRITE

Address offset: 0x1E4

Publish configuration for event WRITE

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

6.20.8.22 PUBLISH_READ

Address offset: 0x1E8

Publish configuration for event READ

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

6.20.8.23 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

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

6.20.8.24 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Enable or disable interrupt for event STOPPED
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	WRITE			Enable or disable interrupt for event WRITE
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	READ			Enable or disable interrupt for event READ
			Disabled	0	Disable
			Enabled	1	Enable

6.20.8.25 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to enable interrupt for event STOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	WRITE			Write '1' to enable interrupt for event WRITE
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	READ			Write '1' to enable interrupt for event READ
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.20.8.26 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				H G F E B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	STOPPED			Write '1' to disable interrupt for event STOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled

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

6.20.8.27 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	Acce	Field	Value ID	Value	Description
A	RW	OVERFLOW			RX buffer overflow detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred
B	RW	DNACK			NACK sent after receiving a data byte
			NotReceived	0	Error did not occur
			Received	1	Error occurred
C	RW	OVERREAD			TX buffer over-read detected, and prevented
			NotDetected	0	Error did not occur
			Detected	1	Error occurred

6.20.8.28 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	Acce	Field	Value ID	Value	Description
A	R	MATCH		[0..1]	Which of the addresses in {ADDRESS} matched the incoming address

6.20.8.29 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	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable TWIS
			Disabled	0	Disable TWIS
			Enabled	9	Enable TWIS

6.20.8.30 PSEL.SCL

Address offset: 0x508

Pin select for SCL signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.20.8.31 PSEL.SDA

Address offset: 0x50C

Pin select for SDA signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.20.8.32 RXD.PTR

Address offset: 0x534

RXD Data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			RXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.20.8.33 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in RXD buffer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in RXD buffer

6.20.8.34 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last RXD transaction

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

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last RXD transaction

6.20.8.35 TXD.PTR

Address offset: 0x544

TXD Data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			TXD Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.20.8.36 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in TXD buffer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in TXD buffer

6.20.8.37 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last TXD transaction

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

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last TXD transaction

6.20.8.38 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	Acce	Field	Value ID	Value	Description
A	RW	ADDRESS			TWI slave address

6.20.8.39 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	Acce	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

6.20.8.40 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	Acce	Field	Value ID	Value	Description
A	RW	ORC			Over-read character. Character sent out in case of an over-read of the transmit buffer.

6.20.9 Electrical specification

6.20.9.1 TWIS slave timing specifications

Symbol	Description	Min.	Typ.	Max.	Units
$f_{TWIS,SCL}$	Bit rates for TWIS[15]	..	..	..	kbps
$t_{TWIS,START}$	Time from PREPARERX/PREPARETX task to ready to receive/ transmit	..	..	..	µs
t_{TWIS,SU_DAT}	Data setup time before positive edge on SCL – all modes	300			ns
t_{TWIS,HD_DAT}	Data hold time after negative edge on SCL – all modes	500			ns
$t_{TWIS,HD\_STA,100kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 100 kbps	5200			ns
$t_{TWIS,HD\_STA,400kbps}$	TWI slave hold time from for START condition (SDA low to SCL low), 400 kbps	1300			ns
$t_{TWIS,SU\_STO,100kbps}$	TWI slave setup time from SCL high to STOP condition, 100 kbps	5200			ns
$t_{TWIS,SU\_STO,400kbps}$	TWI slave setup time from SCL high to STOP condition, 400 kbps	1300			ns
$t_{TWIS,BUF,100kbps}$	TWI slave bus free time between STOP and START conditions, 100 kbps		4700		ns
$t_{TWIS,BUF,400kbps}$	TWI slave bus free time between STOP and START conditions, 400 kbps		1300		ns

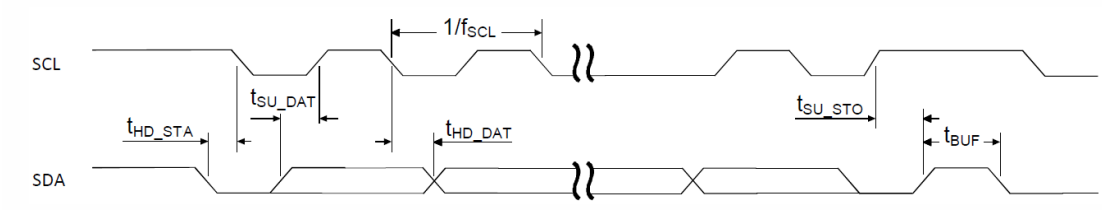

Figure 101: TWIS timing diagram, 1 byte transaction

6.21 UARTE — Universal asynchronous receiver/ transmitter with EasyDMA

The Universal asynchronous receiver/transmitter with EasyDMA (UARTE) offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in hardware at a rate up to 1 Mbps, and EasyDMA data transfer from/to RAM.

Listed here are the main features for UARTE:

- Full-duplex operation
- Automatic hardware flow control
- Optional even parity bit checking and generation
- EasyDMA
- Up to 1 Mbps baudrate
- Return to IDLE between transactions supported (when using HW flow control)
- One or two stop bit

[15] High bit rates or stronger pull-ups may require GPIOs to be set as High Drive, see GPIO chapter for more details.

- Least significant bit (LSB) first

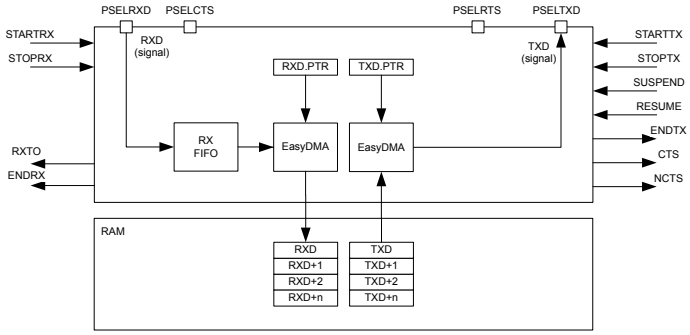

Figure 102: UARTE configuration

The GPIOs used for each UART interface can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pinout and efficient use of board space and signal routing.

6.21.1 EasyDMA

The UARTE implements EasyDMA for reading and writing to and from the RAM.

If the TXD.PTR and the RXD.PTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See Memory on page 20 for more information about the different memory regions.

The .PTR and .MAXCNT registers are double-buffered. They can be updated and prepared for the next RX/TX transmission immediately after having received the RXSTARTED/TXSTARTED event.

The ENDRX/ENDTX event indicates that EasyDMA has finished accessing respectively the RX/TX buffer in RAM.

6.21.2 Transmission

The first step of a DMA transmission is storing bytes in the transmit buffer and configuring EasyDMA. This is achieved by writing the initial address pointer to TXD.PTR, and the number of bytes in the RAM buffer to TXD.MAXCNT. The UARTE transmission is started by triggering the STARTTX task.

After each byte has been sent over the TXD line, a TXDRDY event will be generated.

When all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have been transmitted, the UARTE transmission will end automatically and an ENDTX event will be generated.

A UARTE transmission sequence is stopped by triggering the STOPTX task, a TXSTOPPED event will be generated when the UARTE transmitter has stopped.

If the ENDTX event has not already been generated when the UARTE transmitter has come to a stop, the UARTE will generate the ENDTX event explicitly even though all bytes in the TXD buffer, as specified in the TXD.MAXCNT register, have not been transmitted.

If flow control is enabled through the HWFC field in the CONFIG register, a transmission will be automatically suspended when CTS is deactivated and resumed when CTS is activated again, as illustrated in UARTE transmission on page 331. A byte that is in transmission when CTS is deactivated will be fully transmitted before the transmission is suspended.

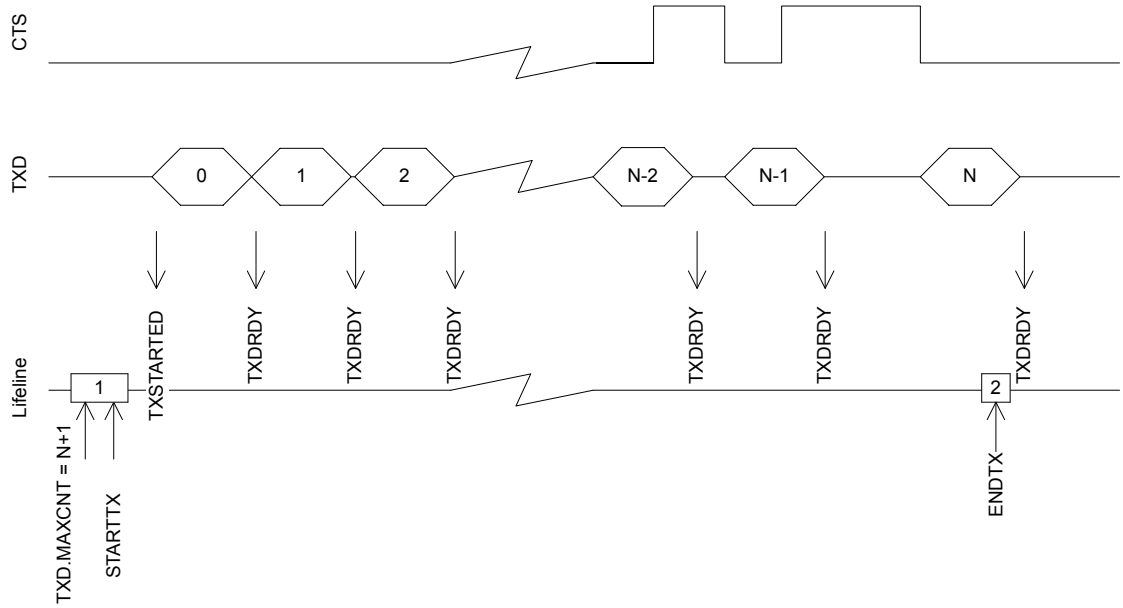

Figure 103: UARTE transmission

The UARTE transmitter will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTTX or after it has been stopped via STOPTX and the TXSTOPPED event has been generated. See POWER — Power control on page 58 for more information about power modes.

6.21.3 Reception

The UARTE receiver is started by triggering the STARTRX task. The UARTE receiver is using EasyDMA to store incoming data in an RX buffer in RAM.

The RX buffer is located at the address specified in the RXD.PTR register. The RXD.PTR register is double-buffered and it can be updated and prepared for the next STARTRX task immediately after the RXSTARTED event is generated. The size of the RX buffer is specified in the RXD.MAXCNT register and the UARTE will generate an ENDRX event when it has filled up the RX buffer, see UARTE reception on page 332.

For each byte received over the RXD line, an RXDRDY event will be generated. This event is likely to occur before the corresponding data has been transferred to Data RAM.

The RXD.AMOUNT register can be queried following an ENDRX event to see how many new bytes have been transferred to the RX buffer in RAM since the previous ENDRX event.

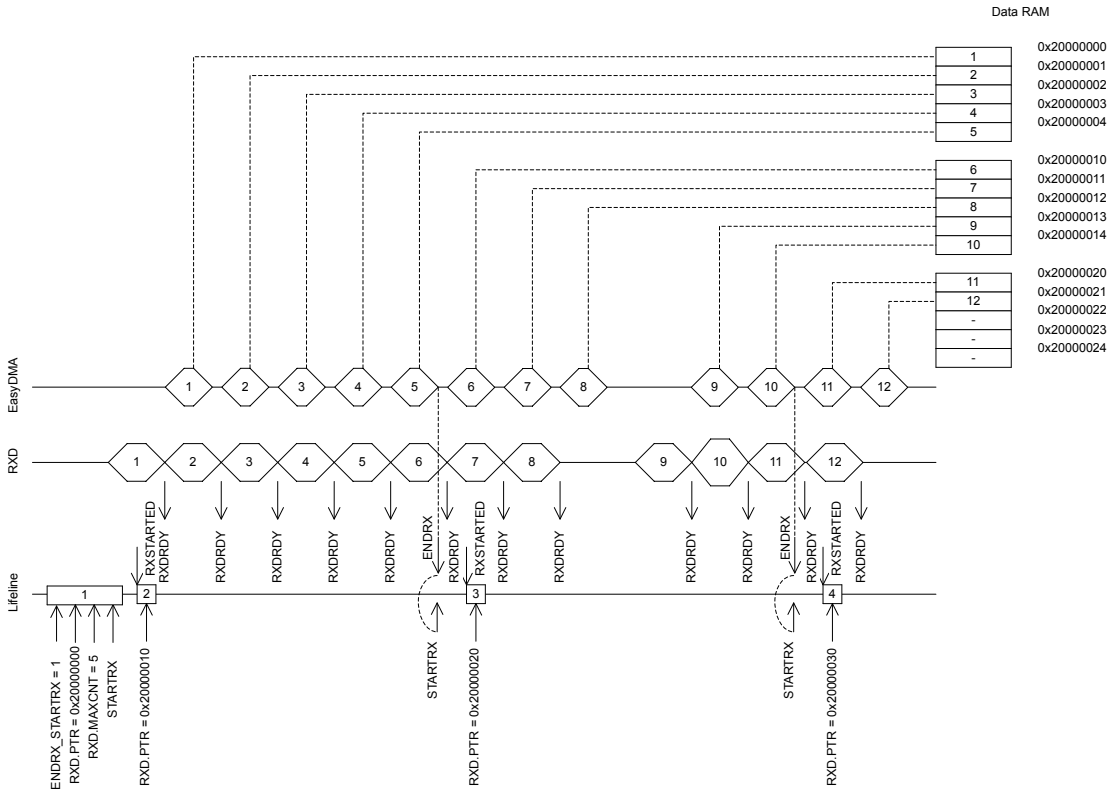

Figure 104: UARTE reception

The UARTE receiver is stopped by triggering the STOPRX task. An RXTO event is generated when the UARTE has stopped. The UARTE will make sure that an impending ENDRX event will be generated before the RXTO event is generated. This means that the UARTE will guarantee that no ENDRX event will be generated after RXTO, unless the UARTE is restarted or a FLUSHRX command is issued after the RXTO event is generated.

Important: If the ENDRX event has not already been generated when the UARTE receiver has come to a stop, which implies that all pending content in the RX FIFO has been moved to the RX buffer, the UARTE will generate the ENDRX event explicitly even though the RX buffer is not full. In this scenario the ENDRX event will be generated before the RXTO event is generated.

To be able to know how many bytes have actually been received into the RX buffer, the CPU can read the RXD.AMOUNT register following the ENDRX event or the RXTO event.

The UARTE is able to receive up to four bytes after the STOPRX task has been triggered as long as these are sent in succession immediately after the RTS signal is deactivated. This is possible because after the RTS is deactivated the UARTE is able to receive bytes for an extended period equal to the time it takes to send 4 bytes on the configured baud rate.

After the RXTO event is generated the internal RX FIFO may still contain data, and to move this data to RAM the FLUSHRX task must be triggered. To make sure that this data does not overwrite data in the RX buffer, the RX buffer should be emptied or the RXD.PTR should be updated before the FLUSHRX task is triggered. To make sure that all data in the RX FIFO is moved to the RX buffer, the RXD.MAXCNT register must be set to RXD.MAXCNT > 4, see UARTE reception with forced stop via STOPRX on page 333. The UARTE will generate the ENDRX event after completing the FLUSHRX task even if the RX FIFO was empty or if the RX buffer does not get filled up. To be able to know how many bytes have actually been received into the RX buffer in this case, the CPU can read the RXD.AMOUNT register following the ENDRX event.

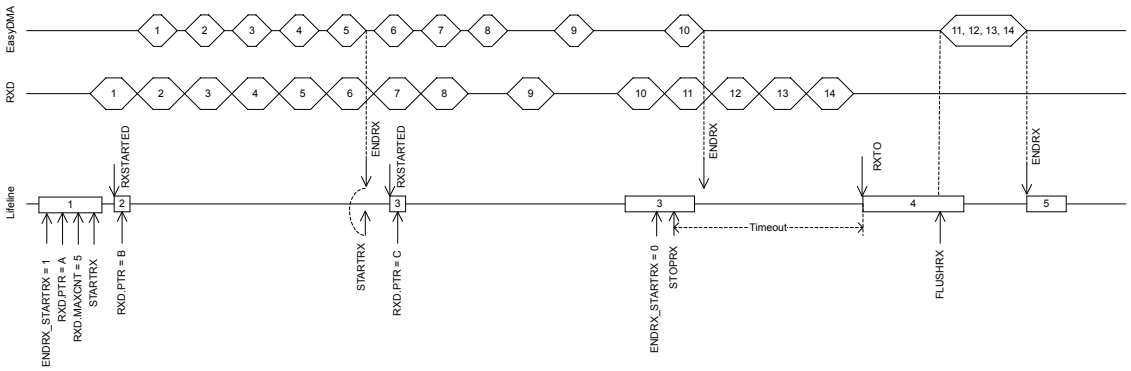

Figure 105: UARTE reception with forced stop via STOPRX

If HW flow control is enabled through the HWFC field in the CONFIG register, the RTS signal will be deactivated when the receiver is stopped via the STOPRX task or when the UARTE is only able to receive four more bytes in its internal RX FIFO.

With flow control disabled, the UARTE will function in the same way as when the flow control is enabled except that the RTS line will not be used. This means that no signal will be generated when the UARTE has reached the point where it is only able to receive four more bytes in its internal RX FIFO. Data received when the internal RX FIFO is filled up, will be lost.

The UARTE receiver will be in its lowest activity level, and consume the least amount of energy, when it is stopped, i.e. before it is started via STARTRX or after it has been stopped via STOPRX and the RXTO event has been generated. See POWER — Power control on page 58 for more information about power modes.

6.21.4 Error conditions

An ERROR event, in the form of a framing error, will be generated if a valid stop bit is not detected in a frame. Another ERROR event, in the form of a break condition, will be generated if the RXD line is held active low for longer than the length of a data frame. Effectively, a framing error is always generated before a break condition occurs.

An ERROR event will not stop reception. If the error was a parity error, the received byte will still be transferred into Data RAM, and so will following incoming bytes. If there was a framing error (wrong stop bit), that specific byte will NOT be stored into Data RAM, but following incoming bytes will.

6.21.5 Using the UARTE without flow control

If flow control is not enabled, the interface will behave as if the CTS and RTS lines are kept active all the time.

6.21.6 Parity and stop bit configuration

When parity is enabled through the PARITY field in the CONFIG register, the parity will be generated automatically from the even parity of TXD and RXD for transmission and reception respectively.

The amount of stop bits can be configured through the STOP field in the CONFIG register.

6.21.7 Low power

When putting the system in low power and the peripheral is not needed, lowest possible power consumption is achieved by stopping, and then disabling the peripheral.

The STOPTX and STOPRX tasks may not be always needed (the peripheral might already be stopped), but if STOPTX and/or STOPRX is sent, software shall wait until the TXSTOPPED and/or RXTO event is received in response, before disabling the peripheral through the ENABLE register.

6.21.8 Pin configuration

The different signals RXD, CTS (Clear To Send, active low), RTS (Request To Send, active low), and TXD associated with the UARTE are mapped to physical pins according to the configuration specified in the PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers respectively.

The PSEL.RXD, PSEL.CTS, PSEL.RTS, and PSEL.TXD registers and their configurations are only used as long as the UARTE is enabled, and retained only for the duration the device is in ON mode. PSEL.RXD, PSEL.RTS, PSEL.RTS and PSEL.TXD must only be configured when the UARTE is disabled.

To secure correct signal levels on the pins by the UARTE when the system is in OFF mode, the pins must be configured in the GPIO peripheral as described in GPIO configuration before enabling peripheral on page 334.

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 91: GPIO configuration before enabling peripheral

6.21.9 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50008000 0x40008000	UARTE	UARTE0 : S UARTE0 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 0	
0x50009000 0x40009000	UARTE	UARTE1 : S UARTE1 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 1	
0x5000A000 0x4000A000	UARTE	UARTE2 : S UARTE2 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 2	
0x5000B000 0x4000B000	UARTE	UARTE3 : S UARTE3 : NS	US	SA	Universal asynchronous receiver/transmitter with EasyDMA 3	

Table 92: Instances

Register	Offset	Security	Description
TASKS_STARTRX	0x000		Start UART receiver
TASKS_STOPRX	0x004		Stop UART receiver
TASKS_STARTTX	0x008		Start UART transmitter
TASKS_STOPTX	0x00C		Stop UART transmitter
TASKS_FLUSHRX	0x02C		Flush RX FIFO into RX buffer
SUBSCRIBE_STARTRX	0x080		Subscribe configuration for task STARTRX
SUBSCRIBE_STOPRX	0x084		Subscribe configuration for task STOPRX
SUBSCRIBE_STARTTX	0x088		Subscribe configuration for task STARTTX
SUBSCRIBE_STOPTX	0x08C		Subscribe configuration for task STOPTX
SUBSCRIBE_FLUSHRX	0x0AC		Subscribe configuration for task FLUSHRX
EVENTS_CTS	0x100		CTS is activated (set low). Clear To Send.

Register	Offset	Security	Description
EVENTS_NCTS	0x104		CTS is deactivated (set high). Not Clear To Send.
EVENTS_RXDRDY	0x108		Data received in RXD (but potentially not yet transferred to Data RAM)
EVENTS_ENDRX	0x110		Receive buffer is filled up
EVENTS_TXDRDY	0x11C		Data sent from TXD
EVENTS_ENDTX	0x120		Last TX byte transmitted
EVENTS_ERROR	0x124		Error detected
EVENTS_RXTO	0x144		Receiver timeout
EVENTS_RXSTARTED	0x14C		UART receiver has started
EVENTS_TXSTARTED	0x150		UART transmitter has started
EVENTS_TXSTOPPED	0x158		Transmitter stopped
PUBLISH_CTS	0x180		Publish configuration for event CTS
PUBLISH_NCTS	0x184		Publish configuration for event NCTS
PUBLISH_RXDRDY	0x188		Publish configuration for event RXDRDY
PUBLISH_ENDRX	0x190		Publish configuration for event ENDRX
PUBLISH_TXDRDY	0x19C		Publish configuration for event TXDRDY
PUBLISH_ENDTX	0x1A0		Publish configuration for event ENDTX
PUBLISH_ERROR	0x1A4		Publish configuration for event ERROR
PUBLISH_RXTO	0x1C4		Publish configuration for event RXTO
PUBLISH_RXSTARTED	0x1CC		Publish configuration for event RXSTARTED
PUBLISH_TXSTARTED	0x1D0		Publish configuration for event TXSTARTED
PUBLISH_TXSTOPPED	0x1D8		Publish configuration for event TXSTOPPED
SHORTS	0x200		Shortcuts between local events and tasks
INTEN	0x300		Enable or disable interrupt
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
ERRORSRC	0x480		Error source Note : this register is read / write one to clear.
ENABLE	0x500		Enable UART
PSEL.RTS	0x508		Pin select for RTS signal
PSEL.TXD	0x50C		Pin select for TXD signal
PSEL.CTS	0x510		Pin select for CTS signal
PSEL.RXD	0x514		Pin select for RXD signal
BAUDRATE	0x524		Baud rate. Accuracy depends on the HFCLK source selected.
RXD.PTR	0x534		Data pointer
RXD.MAXCNT	0x538		Maximum number of bytes in receive buffer
RXD.AMOUNT	0x53C		Number of bytes transferred in the last transaction
TXD.PTR	0x544		Data pointer
TXD.MAXCNT	0x548		Maximum number of bytes in transmit buffer
TXD.AMOUNT	0x54C		Number of bytes transferred in the last transaction
CONFIG	0x56C		Configuration of parity and hardware flow control

Table 93: Register overview

6.21.9.1 TASKS_STARTRX

Address offset: 0x000

Start UART receiver

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTRX			Start UART receiver
			Trigger	1	Trigger task

6.21.9.2 TASKS_STOPRX

Address offset: 0x004

Stop UART receiver

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

6.21.9.3 TASKS_STARTTX

Address offset: 0x008

Start UART transmitter

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_STARTTX			Start UART transmitter
			Trigger	1	Trigger task

6.21.9.4 TASKS_STOPTX

Address offset: 0x00C

Stop UART transmitter

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

6.21.9.5 TASKS_FLUSHRX

Address offset: 0x02C

Flush RX FIFO into RX buffer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_FLUSHRX			Flush RX FIFO into RX buffer
			Trigger	1	Trigger task

6.21.9.6 SUBSCRIBE_STARTRX

Address offset: 0x080

Subscribe configuration for task STARTRX

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

6.21.9.7 SUBSCRIBE_STOPRX

Address offset: 0x084

Subscribe configuration for task STOPRX

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

6.21.9.8 SUBSCRIBE_STARTTX

Address offset: 0x088

Subscribe configuration for task STARTTX

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

6.21.9.9 SUBSCRIBE_STOPTX

Address offset: 0x08C

Subscribe configuration for task STOPTX

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

6.21.9.10 SUBSCRIBE_FLUSHRX

Address offset: 0x0AC

Subscribe configuration for task FLUSHRX

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

6.21.9.11 EVENTS_CTS

Address offset: 0x100

CTS is activated (set low). Clear To Send.

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

6.21.9.12 EVENTS_NCTS

Address offset: 0x104

CTS is deactivated (set high). Not Clear To Send.

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

6.21.9.13 EVENTS_RXDRDY

Address offset: 0x108

Data received in RXD (but potentially not yet transferred to Data RAM)

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

ID	Acce	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

6.21.9.14 EVENTS_ENDRX

Address offset: 0x110

Receive buffer is filled up

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

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDRX			Receive buffer is filled up
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.15 EVENTS_TXDRDY

Address offset: 0x11C

Data sent from TXD

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

ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXDRDY			Data sent from TXD
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.16 EVENTS_ENDTX

Address offset: 0x120

Last TX byte transmitted

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ENDTX			Last TX byte transmitted
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.17 EVENTS_ERROR

Address offset: 0x124

Error detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_ERROR			Error detected
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.18 EVENTS_RXTO

Address offset: 0x144

Receiver timeout

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXTO			Receiver timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.19 EVENTS_RXSTARTED

Address offset: 0x14C

UART receiver has started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_RXSTARTED			UART receiver has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.20 EVENTS_TXSTARTED

Address offset: 0x150

UART transmitter has started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTARTED			UART transmitter has started
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.21 EVENTS_TXSTOPPED

Address offset: 0x158

Transmitter stopped

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TXSTOPPED			Transmitter stopped
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.21.9.22 PUBLISH_CTS

Address offset: 0x180

Publish configuration for event CTS

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

6.21.9.23 PUBLISH_NCTS

Address offset: 0x184

Publish configuration for event NCTS

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

6.21.9.24 PUBLISH_RXDRDY

Address offset: 0x188

Publish configuration for event RXDRDY

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

6.21.9.25 PUBLISH_ENDRX

Address offset: 0x190

Publish configuration for event ENDRX

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

6.21.9.26 PUBLISH_TXDRDY

Address offset: 0x19C

Publish configuration for event TXDRDY

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

6.21.9.27 PUBLISH_ENDTX

Address offset: 0x1A0

Publish configuration for event ENDTX

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ENDTX will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.28 PUBLISH_ERROR

Address offset: 0x1A4

Publish configuration for event ERROR

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event ERROR will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.29 PUBLISH_RXTO

Address offset: 0x1C4

Publish configuration for event RXTO

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXTO will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.30 PUBLISH_RXSTARTED

Address offset: 0x1CC

Publish configuration for event RXSTARTED

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

ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event RXSTARTED will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.21.9.31 PUBLISH_TXSTARTED

Address offset: 0x1D0

Publish configuration for event TXSTARTED

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

6.21.9.32 PUBLISH_TXSTOPPED

Address offset: 0x1D8

Publish configuration for event TXSTOPPED

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

6.21.9.33 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D C
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
C	RW	ENDRX_STARTRX			Shortcut between event ENDRX and task STARTRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut
D	RW	ENDRX_STOPRX			Shortcut between event ENDRX and task STOPRX
			Disabled	0	Disable shortcut
			Enabled	1	Enable shortcut

6.21.9.34 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Enable or disable interrupt for event CTS
			Disabled	0	Disable
			Enabled	1	Enable
B	RW	NCTS			Enable or disable interrupt for event NCTS
			Disabled	0	Disable
			Enabled	1	Enable
C	RW	RXDRDY			Enable or disable interrupt for event RXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
D	RW	ENDRX			Enable or disable interrupt for event ENDRX
			Disabled	0	Disable
			Enabled	1	Enable
E	RW	TXDRDY			Enable or disable interrupt for event TXDRDY
			Disabled	0	Disable
			Enabled	1	Enable
F	RW	ENDTX			Enable or disable interrupt for event ENDTX
			Disabled	0	Disable
			Enabled	1	Enable
G	RW	ERROR			Enable or disable interrupt for event ERROR
			Disabled	0	Disable
			Enabled	1	Enable
H	RW	RXTO			Enable or disable interrupt for event RXTO
			Disabled	0	Disable
			Enabled	1	Enable
I	RW	RXSTARTED			Enable or disable interrupt for event RXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
J	RW	TXSTARTED			Enable or disable interrupt for event TXSTARTED
			Disabled	0	Disable
			Enabled	1	Enable
L	RW	TXSTOPPED			Enable or disable interrupt for event TXSTOPPED
			Disabled	0	Disable
			Enabled	1	Enable

6.21.9.35 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to enable interrupt for event CTS
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to enable interrupt for event NCTS
			Set	1	Enable

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to enable interrupt for event RXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to enable interrupt for event ENDRX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to enable interrupt for event TXDRDY
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to enable interrupt for event ENDTX
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to enable interrupt for event ERROR
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to enable interrupt for event RXTO
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to enable interrupt for event RXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to enable interrupt for event TXSTARTED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to enable interrupt for event TXSTOPPED
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.9.36 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				L J I H G F E D C B A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CTS			Write '1' to disable interrupt for event CTS

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

ID	Acce	Field	Value ID	Value	Description
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
B	RW	NCTS			Write '1' to disable interrupt for event NCTS
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
C	RW	RXDRDY			Write '1' to disable interrupt for event RXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
D	RW	ENDRX			Write '1' to disable interrupt for event ENDRX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
E	RW	TXDRDY			Write '1' to disable interrupt for event TXDRDY
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
F	RW	ENDTX			Write '1' to disable interrupt for event ENDTX
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
G	RW	ERROR			Write '1' to disable interrupt for event ERROR
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
H	RW	RXTO			Write '1' to disable interrupt for event RXTO
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
I	RW	RXSTARTED			Write '1' to disable interrupt for event RXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
J	RW	TXSTARTED			Write '1' to disable interrupt for event TXSTARTED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled
L	RW	TXSTOPPED			Write '1' to disable interrupt for event TXSTOPPED
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.21.9.37 ERRORSRC

Address offset: 0x480

Error source

Note : 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	Acce	Field	Value ID	Value	Description
A	RW	OVERRUN			Overrun error A start bit is received while the previous data still lies in RXD. (Previous data is lost.)
			NotPresent	0	Read: error not present
			Present	1	Read: error present
B	RW	PARITY			Parity error A character with bad parity is received, if HW parity check is enabled.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
C	RW	FRAMING			Framing error occurred A valid stop bit is not detected on the serial data input after all bits in a character have been received.
			NotPresent	0	Read: error not present
			Present	1	Read: error present
D	RW	BREAK			Break condition The serial data input is '0' for longer than the length of a data frame. (The data frame length is 10 bits without parity bit, and 11 bits with parity bit.).
			NotPresent	0	Read: error not present
			Present	1	Read: error present

6.21.9.38 ENABLE

Address offset: 0x500

Enable UART

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				A A A A
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	ENABLE			Enable or disable UARTE
			Disabled	0	Disable UARTE
			Enabled	8	Enable UARTE

6.21.9.39 PSEL.RTS

Address offset: 0x508

Pin select for RTS signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.40 PSEL.TXD

Address offset: 0x50C

Pin select for TXD signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.41 PSEL.CTS

Address offset: 0x510

Pin select for CTS signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.42 PSEL.RXD

Address offset: 0x514

Pin select for RXD signal

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			C 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	Acce	Field	Value ID	Value	Description
A	RW	PIN		[0..31]	Pin number
C	RW	CONNECT			Connection
			Disconnected	1	Disconnect
			Connected	0	Connect

6.21.9.43 BAUDRATE

Address offset: 0x524

Baud rate. Accuracy depends on the HFCLK source selected.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A
Reset 0x04000000			**0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	BAUDRATE			Baud rate
			Baud1200	0x0004F000	1200 baud (actual rate: 1205)
			Baud2400	0x0009D000	2400 baud (actual rate: 2396)
			Baud4800	0x0013B000	4800 baud (actual rate: 4808)
			Baud9600	0x00275000	9600 baud (actual rate: 9598)
			Baud14400	0x003AF000	14400 baud (actual rate: 14401)
			Baud19200	0x004EA000	19200 baud (actual rate: 19208)
			Baud28800	0x0075C000	28800 baud (actual rate: 28777)
			Baud31250	0x00800000	31250 baud
			Baud38400	0x009D0000	38400 baud (actual rate: 38369)
			Baud56000	0x00E50000	56000 baud (actual rate: 55944)
			Baud57600	0x00EB0000	57600 baud (actual rate: 57554)
			Baud76800	0x013A9000	76800 baud (actual rate: 76923)
			Baud115200	0x01D60000	115200 baud (actual rate: 115108)
			Baud230400	0x03B00000	230400 baud (actual rate: 231884)
			Baud250000	0x04000000	250000 baud
			Baud460800	0x07400000	460800 baud (actual rate: 457143)
			Baud921600	0x0F000000	921600 baud (actual rate: 941176)
			Baud1M	0x10000000	1Mega baud

6.21.9.44 RXD.PTR

Address offset: 0x534

Data pointer

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

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.21.9.45 RXD.MAXCNT

Address offset: 0x538

Maximum number of bytes in receive buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in receive buffer

6.21.9.46 RXD.AMOUNT

Address offset: 0x53C

Number of bytes transferred in the last transaction

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.21.9.47 TXD.PTR

Address offset: 0x544

Data pointer

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

ID	Acce	Field	Value ID	Value	Description
A	RW	PTR			Data pointer

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.21.9.48 TXD.MAXCNT

Address offset: 0x548

Maximum number of bytes in transmit buffer

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	A A A A A A A A A A A A A A A A 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	Acce	Field	Value ID	Value	Description
A	RW	MAXCNT		[1..0x1FFF]	Maximum number of bytes in transmit buffer

6.21.9.49 TXD.AMOUNT

Address offset: 0x54C

Number of bytes transferred in the last transaction

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

ID	Acce	Field	Value ID	Value	Description
A	R	AMOUNT		[1..0x1FFF]	Number of bytes transferred in the last transaction

6.21.9.50 CONFIG

Address offset: 0x56C

Configuration of parity and hardware flow control

Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID	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	Acce	Field	Value ID	Value	Description
A	RW	HWFC			Hardware flow control
			Disabled	0	Disabled
			Enabled	1	Enabled
B	RW	PARITY			Parity
			Excluded	0x0	Exclude parity bit
			Included	0x7	Include even parity bit
C	RW	STOP			Stop bits
			One	0	One stop bit
			Two	1	Two stop bits

6.21.10 Electrical specification

6.21.10.1 UARTE electrical specification

Symbol	Description	Min.	Typ.	Max.	Units
f_{UARTE}	Baud rate for UARTE[16].			1000	kbps
$t_{UARTE,CTSH}$	CTS high time	1			µs
$t_{UARTE,START}$	Time from STARTRX/STARTTX task to transmission started	..	..	..	µs

6.22 WDT — Watchdog timer

A countdown watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up.

The watchdog timer is started by triggering the START task.

The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU. The watchdog is implemented as a down-counter that generates a TIMEOUT event when it wraps over after counting down to 0. When the watchdog timer is started through the START task, the watchdog counter is loaded with the value specified in the CRV register. This counter is also reloaded with the value specified in the CRV register when a reload request is granted.

[16] High baud rates may require GPIOs to be set as High Drive, see GPIO chapter for more details.

The watchdog's timeout period is given by:

```
timeout [s] = ( CRV + 1 ) / 32768
```

When started, the watchdog will automatically force the 32.768 kHz RC oscillator on as long as no other 32.768 kHz clock source is running and generating the 32.768 kHz system clock, see chapter CLOCK — Clock control on page 64.

6.22.1 Reload criteria

The watchdog has eight separate reload request registers, which shall be used to request the watchdog to reload its counter with the value specified in the CRV register. To reload the watchdog counter, the special value 0x6E524635 needs to be written to all enabled reload registers.

One or more RR registers can be individually enabled through the RREN register.

6.22.2 Temporarily pausing the watchdog

By default, the watchdog will be active counting down the down-counter while the CPU is sleeping and when it is halted by the debugger. It is however possible to configure the watchdog to automatically pause while the CPU is sleeping as well as when it is halted by the debugger.

6.22.3 Watchdog reset

A TIMEOUT event will automatically lead to a watchdog reset.

See Reset on page 54 for more information about reset sources. If the watchdog is configured to generate an interrupt on the TIMEOUT event, the watchdog reset will be postponed with two 32.768 kHz clock cycles after the TIMEOUT event has been generated. Once the TIMEOUT event has been generated, the impending watchdog reset will always be effectuated.

The watchdog must be configured before it is started. After it is started, the watchdog's configuration registers, which comprise registers CRV, RREN, and CONFIG, will be blocked for further configuration.

The watchdog can be reset from several reset sources, see Reset behavior on page 55.

When the device starts running again, after a reset, or waking up from OFF mode, the watchdog configuration registers will be available for configuration again.

6.22.4 Registers

Base address	Peripheral	Instance	Secure mapping	DMA security	Description	Configuration
0x50018000	WDT	WDT : S	US	NA	Watchdog timer	
0x40018000		WDT : NS				

Table 94: Instances

Register	Offset	Security	Description
TASKS_START	0x000		Start the watchdog
SUBSCRIBE_START	0x080		Subscribe configuration for task START
EVENTS_TIMEOUT	0x100		Watchdog timeout
PUBLISH_TIMEOUT	0x180		Publish configuration for event TIMEOUT
INTENSET	0x304		Enable interrupt
INTENCLR	0x308		Disable interrupt
RUNSTATUS	0x400		Run status

Register	Offset	Security	Description
REQSTATUS	0x404		Request status
CRV	0x504		Counter reload value
RREN	0x508		Enable register for reload request registers
CONFIG	0x50C		Configuration register
RR[0]	0x600		Reload request 0
RR[1]	0x604		Reload request 1
RR[2]	0x608		Reload request 2
RR[3]	0x60C		Reload request 3
RR[4]	0x610		Reload request 4
RR[5]	0x614		Reload request 5
RR[6]	0x618		Reload request 6
RR[7]	0x61C		Reload request 7

Table 95: Register overview

6.22.4.1 TASKS_START

Address offset: 0x000

Start the watchdog

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	TASKS_START			Start the watchdog
			Trigger	1	Trigger task

6.22.4.2 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
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that task START will subscribe to
B	RW	EN			
			Disabled	0	Disable subscription
			Enabled	1	Enable subscription

6.22.4.3 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	Acce	Field	Value ID	Value	Description
A	RW	EVENTS_TIMEOUT			Watchdog timeout
			NotGenerated	0	Event not generated
			Generated	1	Event generated

6.22.4.4 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
Reset 0x00000000				**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	CHIDX		[15..0]	Channel that event TIMEOUT will publish to.
B	RW	EN			
			Disabled	0	Disable publishing
			Enabled	1	Enable publishing

6.22.4.5 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	Acce	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to enable interrupt for event TIMEOUT
			Set	1	Enable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.4.6 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	Acce	Field	Value ID	Value	Description
A	RW	TIMEOUT			Write '1' to disable interrupt for event TIMEOUT
			Clear	1	Disable
			Disabled	0	Read: Disabled
			Enabled	1	Read: Enabled

6.22.4.7 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	Acce	Field	Value ID	Value	Description
A	R	RUNSTATUSWDT			Indicates whether or not the watchdog is running
			NotRunning	0	Watchdog not running
			Running	1	Watchdog is running

6.22.4.8 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	Acce	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

6.22.4.9 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	Acce	Field	Value ID	Value	Description
A	RW	CRV		[0x0000000F..0xFFFFFFFF]	Counter reload value in number of cycles of the 32.768 kHz clock

6.22.4.10 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	Acce	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

6.22.4.11 CONFIG

Address offset: 0x50C

Configuration register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				C A
Reset 0x00000001				**0 0 0 0 0 0 0 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	Acce	Field	Value ID	Value	Description
A	RW	SLEEP			Configure the watchdog to either be paused, or kept running, while the CPU is sleeping
			Pause	0	Pause watchdog while the CPU is sleeping
			Run	1	Keep the watchdog running while the CPU is sleeping
C	RW	HALT			Configure the watchdog to either be paused, or kept running, while the CPU is halted by the debugger
			Pause	0	Pause watchdog while the CPU is halted by the debugger
			Run	1	Keep the watchdog running while the CPU is halted by the debugger

6.22.4.12 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	Acce	Field	Value ID	Value	Description
A	W	RR			Reload request register
			Reload	0x6E524635	Value to request a reload of the watchdog timer

6.22.5 Electrical specification

6.22.5.1 Watchdog Timer Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{WDT}	Time out interval	31 µs		36 h	

7 LTE modem

7.1 Introduction

The long term evolution (LTE) modem consists of baseband processing and RF parts, which together implement a complete 3GPP LTE release 13 (Rel-13) Cat-M1 and Cat-NB1 and LTE release 14 (Rel-14) Cat-NB1 and Cat-NB2 capable product.

As illustrated in the image below, the following is a part of the LTE modem:

- RF transceiver
- Modem baseband (BB)
- Embedded flash/RAM
- Modem host processor and peripherals

The modem baseband and host processor provide functions for the LTE L1, L2 and L3 (layer 1, 2 and 3 respectively) as well as IP communication layers. Modem peripherals provide hardware services for modem operating system and for modem secure execution environment.

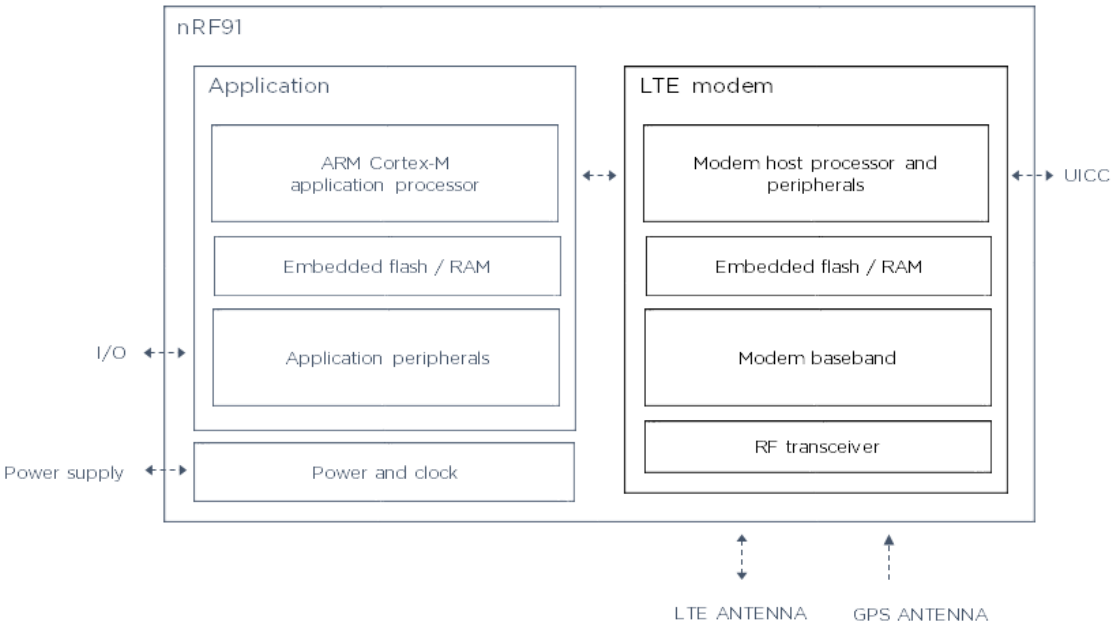

Figure 106: LTE modem within the nRF91

Application and modem domains are interacting through interprocessor communication (IPC) mechanism. LTE modem is accessible to user through the modem API.

The application processor is the master in the system and responsible for starting and stopping of the modem. LTE modem enables the clocks and power required for its own operation. Shared resources, such as e.g. clocks, are handled within the platform and require no user involvement. In cases where a hard fault is detected in the modem, the application domain will perform a hard reset for the modem.

Note: For details regarding the modem API, please refer to *nRF Connect SDK* document and *nRF91 AT Commands, Command Reference Guide* document.

Key features of the LTE modem are:

- Complete modem with baseband and RF transceiver
- 3GPP release 13 compliant LTE categories:
  - Cat-M1 (eMTC - enhanced machine type communication)
  - Cat-NB1 (NB-IoT - narrowband internet of things (IoT))
- 3GPP release 14 compliant LTE categories:
  - Cat-NB1 (NB-IoT)
  - Cat-NB2 (NB-IoT)
- Power saving modes
- Supporting LTE bands from 700 MHz to 2.2 GHz through a single typical 50 Ω antenna pin.
- RX sensitivity: -108 dBm for Cat-M1 and -114 dBm for Cat-NB1 and Cat-NB2
  - As defined in 3GPP conformance test specification TS 36.521-1
- 1.8 V MIPI RFFE (RF front-end) digital control interface and MAGPIO control interface for external RF applications.
- 1.8 V UICC (universal integrated circuit card) interface, based on ISO/IEC 7816-3 and compliant with:
  - ICC (ETSI TS 102 221)
  - eUICC (ETSI TS 103 383)

Note: nRF9160 is able to run different modem FW builds that define the final modem feature set in a specific nRF9160 based application.

7.2 SIM card interface

LTE modem supports the UICC (universal integrated circuit card) interface.

Only the UICCs with the electrical interface specified in ISO/IEC 7816-3 are supported, meaning that the UICCs with IC-USB, CLF or MMC interfaces are not supported.

The supported UICC/eUICC interface is compliant with:

- ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics
- ETSI TS 103 383: Smart Cards; Embedded UICC; Requirements Specification

The physical interface towards the eUICC is the same as towards the removable UICC.

Only the class C (supply voltage 1.8 V nominal) operation is supported. Support for the legacy class B (supply voltage 3.0 V nominal) operation must be built with external components, including the external power supply and the level shifters towards the LTE modem UICC interface.

LTE modem controls the physical interfaces towards the UICC and implements the transport protocol over the four-pin ISO/IEC 7816-3 interface:

- VCC (power supply): LTE modem drives this
- CLK (clock signal): LTE modem drives this
- RST (reset signal): LTE modem drives this

- I/O (input/output serial data): Bi-directional

The interface and the connections between LTE modem, UICC connector, and the ESD device is shown in the figure below.

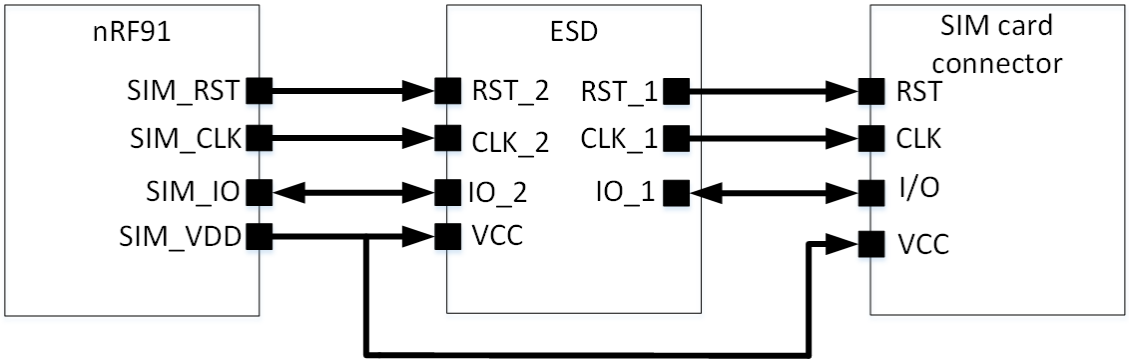

Figure 107: Connections between LTE modem, card connector, and the ESD device

Only standard transmission speeds are supported as specified in ETSI TS 102 221.

Important: LTE modem must be stopped through the modem API, before removing the UICC.

An ESD (electrostatic discharge) protection device compatible with UICC cards must be used between the removable card and the LTE modem, to protect LTE modem against a harmful electrostatic discharge from the card connector.

7.3 LTE modem coexistence interface

LTE modem uses a dedicated three-pin interface for RF interference avoidance towards a companion radio device e.g. an external Bluetooth® Low Energy device.

The inputs and outputs for this interface:

- COEX0: Input to the LTE modem from the external device. When active high, indicates that the external device transceiver is turned on.
- COEX1: Output from the LTE modem to the external device. Active high time mark pulse, which is synchronous to LTE system time.
- COEX2: Output from the LTE modem to the external device. When active high, indicates that the LTE modem transceiver is turned on.
  - COEX2 can also be treated as active low grant from LTE modem to the external device, indicating grant to transmit.
  - **Note:** COEX2 pin requires an external pull-down resistor in 100 kΩ size range to be used.

Simultaneous receiving by LTE modem and external device is always possible, and by so means no coexistence signaling needed when only receiving is done on the external device side.

7.4 LTE modem RF control external interface

LTE modem provides dedicated 1.8 V digital interfaces for controlling external RF applications, such as antenna tuner devices:

- MIPI RFFE interface pins: VIO, SCLK, SDATA.
- MAGPIO interface pins: MAGPIO0, MAGPIO1, MAGPIO2.

LTE modem drives these outputs timing accurately according to LTE protocol timing to set e.g. the correct antenna tuner settings per used frequency.

User needs to inform the LTE modem through the modem API about the particular RF application e.g. antenna tuner device configuration, so that LTE modem knows how to drive it.

Note: For details regarding the modem API and supported RF external control features, please refer to *nRF91 AT Commands, Command Reference Guide* document.

7.5 RF front-end interface

nRF9160 has a single-ended (SE) 50 Ω antenna interface to connect directly to antenna.

7.6 Electrical specification

7.6.1 Key RF parameters for Cat-M1

Note: The bands listed in this table define the certified bands.

Symbol	Description	Min.	Typ.	Max.	Units
Supported LTE	Supported LTE standards		LTE Rel-13 Cat-M1 HD-FDD		
Bands supported	Certified bands supported		USA and Canada: B4, B13. Europe: B3, B20.		
Transmission bandwidth	Maximum bandwidth		1.4		MHz

7.6.2 Key RF parameters for Cat-NB1 and Cat-NB2

Note: The bands listed in this table define the certified bands.

Symbol	Description	Min.	Typ.	Max.	Units
Supported LTE	Supported LTE standards		LTE Rel-13 Cat-NB1 HD-FDD, LTE Rel-14 Cat-NB1 and Cat-NB2 HD-FDD		
Bands supported	Certified bands supported		For Cat-NB1, Europe: B3, B20		
Transmission bandwidth	Maximum bandwidth		200		kHz

7.6.3 Receiver parameters for Cat-M1

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_RX}$	RX operation frequency range at ANT (pin 61)	729		2200	MHz
Z_{in}	Input impedance, single-ended		50		Ω
Sensitivity, low band	LTE 1.4 MHz without coverage extension	-103	-108		dBm
Sensitivity, mid band	LTE 1.4 MHz without coverage extension	-103	-107		dBm

7.6.4 Receiver parameters for Cat-NB1 and Cat-NB2

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_RX}$	RX operation frequency range at ANT (pin 61)	729		2200	MHz
Z_{in}	Input impedance, single-ended		50		Ω
Sensitivity, low band	NB 200 kHz without coverage extension	-108	-114		dBm
Sensitivity, mid band	NB 200 kHz without coverage extension	-108	-113		dBm

7.6.5 Transmitter parameters for Cat-M1

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_TX}$	TX operation frequency range at ANT (pin 61)	699		1910	MHz
Z_{out}	Output impedance, single-ended		50		Ω
Maximum output power	Maximum output power, 3GPP specification		23		dBm
Minimum output power	Minimum output power, 3GPP specification		-40		dBm
Pout maximum accuracy	Pout maximum accuracy, internal specification		+-2		dB

7.6.6 Transmitter parameters for Cat-NB1 and Cat-NB2

Symbol	Description	Min.	Typ.	Max.	Units
$Freq_{range\_ANT\_TX}$	TX operation frequency range at ANT (pin 61)	699		1910	MHz
Z_{out}	Output impedance, single-ended		50		Ω
Maximum output power	Maximum output power, 3GPP specification		23		dBm
Minimum output power	Minimum output power, 3GPP specification		-40		dBm
Pout maximum accuracy	>Pout maximum accuracy, Internal specification		+-2		dB

8 GPS receiver

The GPS receiver supports GPS L1C/A reception. Operation is time multiplexed with LTE modem, and it is possible to obtain the GPS position while LTE is in DRX or PSM mode.

GPS receiver is visible to user only through GPS API.

Key features of the GPS receiver are:

- GPS L1C/A is supported
- Modes of operation:
  - Single shot (cold start mode by default)
  - Position fix per fixed interval, e.g. 2 minutes (starts with cold start, sequential fixes with hot start)
- Power saving modes
- Antenna interface options:
  - Dedicated GPS antenna or shared antenna with LTE M1/NB1
  - With or without external low-noise amplifier (LNA)
- Performance:
  - Acquisition sensitivity: -144 dBm for cold start, -147 dBm for hot start
  - Acquisition time: 30 seconds for cold start, 5 seconds for hot start
  - Tracking sensitivity: -149 dBm (open sky received power ≥ -130 dBm)
  - Accuracy: 5 m

9 Debug and trace

9.1 Overview

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

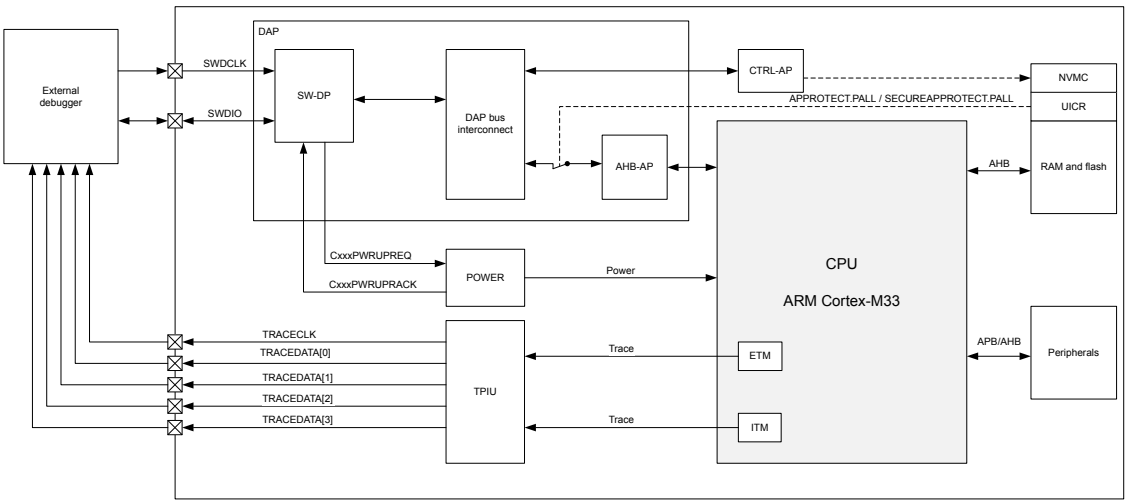

Figure 108: Debug and trace overview

The main features of the debug and trace system are:

- Two-pin serial wire debug (SWD) interface, protocol version 1
- Access port connection
  - Breakpoint unit (BPU) supports eight hardware breakpoint comparators
  - Data watchpoint and trace (DWT) unit supports four watchpoint comparators
  - Instrumentation trace macrocell (ITM)
  - Embedded trace macrocell (ETM)
  - Access protection through APPROTECT, ERASEPROTECT and SECUREAPPROTECT
- Trace port interface unit (TPUI)
  - 4-bit parallel trace of ITM and ETM trace data

Note: When a system contains multiple CPU domains, it is important to notice that if one domain (subsystem A) has master rights on another domain (subsystem B), the master subsystem can have access to data from the slave subsytem. In this example, even if subsystem B is locked by APPROTECT or ERASEPROTECT, subsystem A can access some data for subsystem B. Consequently, even if the security permissions are managed per subsystem, it is mandatory to have a global approach to the protection. Protecting a slave subsystem does not guarantee system security if the master subsystem is not protected.

9.1.1 Special consideration regarding debugger access

A debugger can be restricted to debug non-secure code only, and access non-secure memory regions and peripherals using register SECUREAPPROTECT on page 43. Register APPROTECT on page 42 will block all debugger access.

Debugger accesses are controlled as described in table below.

Debugging capability	UICR.APPROTECT.PALL	UICR.SECUREAPPROTECT.PALL
Secure and non-secure code	Unprotected	Unprotected
Non-secure code only	Unprotected	Protected
No debugging possible	Protected	-

Table 96: Debugger access control

If a RAM or flash region has its permission set to allow code execution, the content of this region will be visible to the debugger even if the read permission is not set. This allows a debugger to display the content of the code being executed.

9.1.2 DAP - Debug access port

An external debugger can access the device via the debug access port (DAP).

The DAP implements a standard ARM® CoreSight™ serial wire debug port (SW-DP). The SW-DP implements the serial wire debug (SWD) protocol that is a two-pin serial interface, see SWDCLK and SWDIO illustrated in figure Debug and trace overview on page 365.

In addition to the default access port in the application CPU (AHB-AP), the DAP includes a custom control access port (CTRL-AP). The CTRL-AP is described in more detail in CTRL-AP - Control access port on page 368.

Note:

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

There are several access ports that connect to different parts of the system. An overview is given in the table below.

AP ID	Type	Description
0	AHB-AP	Application subsystem access port
4	CTRL-AP	Application subsystem control access port

Table 97: Access port overview

The AHB-AP and APB-AP are standard ARM® components, and documented in *ARM CoreSight SoC-400 Technical Reference Manual, revision r3p2*. The control access port (CTRL-AP) is proprietary, and described in more detail in CTRL-AP - Control access port on page 368.

9.1.3 Debug interface mode

Before the external debugger can access the CPU's access port (AHB-AP) or the control access port (CTRL-AP), the debugger must first request the device to power up via CxxxPWRUPREQ in the SWJ-DP.

As long as the debugger is requesting power via CxxxPWRUPREQ, the device will be in debug interface mode. Otherwise, the device is in normal mode. When a debug session is over, the external debugger must make sure to put the device back into normal mode and then a pin reset should be performed. The reason is that the overall power consumption is higher in debug interface mode compared to normal mode.

Some peripherals behave differently in debug interface mode compared to normal mode. The differences are described in more detail in the chapters of the peripherals that are affected.

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

If the device is in System OFF when power is requested via CxxxPWRUPREQ, the system will wake up and the DIF flag in RESETREAS on page 63 will be set.

9.1.4 Real-time debug

The device supports real-time debugging, which allows interrupts to execute to completion in real time when breakpoints are set in thread mode or lower priority interrupts.

Real-time debugging thus enables the developer to set a breakpoint and single-step through their code without a failure of the real-time event-driven threads running at higher priority. For example, this enables the device to continue to service the high-priority interrupts of an external controller or sensor without failure or loss of state synchronization while the developer steps through code in a low-priority thread.

9.1.5 Trace

The device supports ETM and ITM trace.

Trace data from the ETM and the ITM is sent to an external debugger via a 4-bit wide parallel trace port (TPIU), see TRACEDATA[0] through TRACEDATA[3], and TRACECLK in Debug and trace overview on page 365.

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

TPIU's trace pins are multiplexed with GPIOs, see Pin assignments on page 379 for more information.

Trace speed is configured in the TRACEPORTSPEED on page 378 register. The speed of the trace pins depends on the DRIVE setting of the GPIOs that the trace pins are multiplexed with. Only S0S1 and H0H1 drives are suitable for debugging. S0S1 is the default DRIVE at reset. If parallel or serial trace port signals are not fast enough in the debugging conditions, all GPIOs in use for tracing should be set to high drive (H0H1). The user shall make sure that DRIVE setting for these GPIOs is not overwritten by software during the debugging session.

9.1.6 Registers

Register	Offset	Security	Description
TARGETID	0x042		The TARGETID register provides information about the target when the host is connected to a single device. The TARGETID register is accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x2.

Table 98: Register overview

9.1.6.1 TARGETID

Address offset: 0x042

The TARGETID register provides information about the target when the host is connected to a single device.

The TARGETID register is accessed by a read of DP register 0x4 when the DPBANKSEL bit in the SELECT register is set to 0x2.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID				D D D D C C C C C C C C C C C C B B B B B B B B B B A
Reset 0x10090289				**0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1**
ID	Acce	Field	Value ID	Value	Description
A	R	UNUSED			Reserved, read-as-one
B	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
C	R	TPARTNO			Part number
			nRF91	9	nRF91 Series
D	R	TREVISION			Target revision
			<keyword keyref="devicename" />	1	nRF9160

9.1.7 Electrical specification

9.1.7.1 Trace port

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

9.2 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 in the debug access port (DAP) are disabled by the access port protection.

For overview of other access ports in DAP, see DAP - Debug access port on page 366.

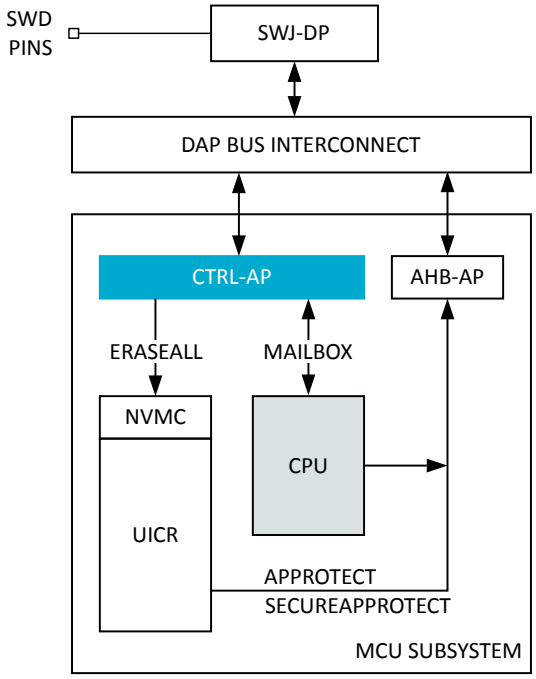

Figure 109: Control access port details

Access port protection (APPROTECT) blocks the debugger access to the AHB-AP, and prevents read and write access to all CPU registers and memory-mapped addresses. It is possible to enable access port protection for both secure and non-secure mode, using registers UICR.SECUREAPPROTECT and UICR.APPROTECT respectively. The debugger can use register APPROTECT.STATUS on page 372 to read the status of secure and non-secure access port protection.

Control access port has the following features:

- Soft reset
- Erase all
- Mailbox interface
- Debug of protected devices

9.2.1 Reset request

The debugger can request the device to perform a soft reset.

Register RESET on page 371 is used to request the soft reset. Once the soft reset is performed, the reset reason is accessible to on-chip firmware through register . For more information about the soft reset, see Reset on page 54.

9.2.2 Erase all

Erase all function gives debugger the possibility of triggering an erase of flash, user information configuration registers (UICR), RAM, including all peripheral settings, as well as removing the access port protection.

To trigger an erase all function, the debugger can write the register ERASEALL on page 371. Register ERASEALLSTATUS on page 371 will read as busy for the duration of the operation. After the next reset, the access port protection is removed.

For slave MCU's, the ERASEALL command will also affect the application MCU. The ERASEALL command is performed on the application MCU first, independently of how the application is protected, and then on the slave MCU.

Erase all protection

It is possible to prevent debugger from performing an erase all operation by writing to register ERASEPROTECT on page 43. Once the register is configured and the device reset, the control access port ERASEALL operation is disabled, and all flash write and erase operations are restricted to firmware. In addition, it is still possible to write/erase from debugger as long as APPROTECT on page 42 is not set.

Note: Setting ERASEPROTECT on page 43 has no effect on debugger access, only on erase all operation.

Register ERASEPROTECT.STATUS on page 372 holds the status for erase protection.

9.2.3 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 373 with its corresponding status register MAILBOX.TXSTATUS on page 373, and a receive register MAILBOX.RXDATA on page 373 with its corresponding status register MAILBOX.RXSTATUS on page 373. 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.

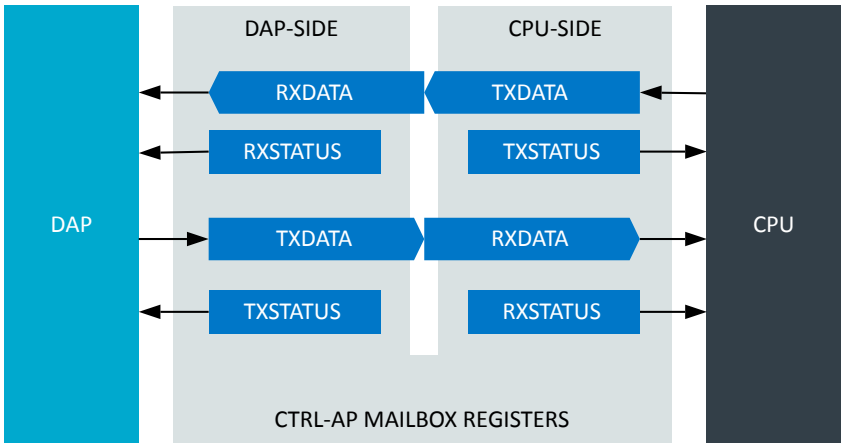

Figure 110: Mailbox register interface

Mailbox transfer sequence

1. Sender writes TXDATA
2. Hardware sets sender's TXSTATUS to DataPending
3. Hardware sets receiver's RXSTATUS to DataPending
4. Receiver reads RXDATA
5. Hardware sets receiver's RXSTATUS to NoDataPending
6. Hardware sets sender's TXSTATUS to NoDataPending

9.2.4 Unlocking of access port

The access port protection mechanisms can be temporarily bypassed to erase or debug the device.

Note: The mailbox feature of the CTRL-AP can be used by firmware to authenticate the debugger before allowing it to use the access port.

Disabling the erase all protection

To bypass ERASEPROTECT setting, making it possible for the access port to erase all memories, both the debugger and firmware must set the ERASEALL field in their respective ERASEPROTECTDISABLE registers. As soon as both registers have been written, the device is automatically erased using erase all function as described in Erase all on page 369, and then the access port is made available.

Note: To prevent misuse, the write-once register ERASEPROTECT.DISABLE on page 375 should be set to Default as early in the start-up process as possible. Once written, it will not be possible to remove the erase protection until next reset.

9.2.5 Registers

Register	Offset	Security	Description
RESET	0x000		Soft reset request.
ERASEALL	0x004		Perform a secure erase of the device. The device will be returned to factory default settings upon next reset.
ERASEALLSTATUS	0x008		Status register for the ERASEALL operation
APPROTECT.STATUS	0x00C		Status register for access port protection
ERASEPROTECT.STATUS	0x018		Status register for UICR ERASEPROTECT configuration
ERASEPROTECT.DISABLE	0x01C		Unlock ERASEPROTECT and perform ERASEALL

Register	Offset	Security	Description
MAILBOX.TXDATA	0x020		Data sent from the debugger to the CPU
MAILBOX.TXSTATUS	0x024		Status to indicate if data sent from the debugger to the CPU has been read
MAILBOX.RXDATA	0x028		Data sent from the CPU to the debugger
MAILBOX.RXSTATUS	0x02C		Status to indicate if data sent from the CPU to the debugger has been read
IDR	0x0FC		CTRL-AP Identification Register, IDR

Table 99: Register overview

9.2.5.1 RESET

Address offset: 0x000

Soft reset request.

This register is automatically deactivated by writing Erase to ERASEALL, it is then kept inactive until a reset source affecting the debug system is asserted. See Reset behavior on page 55.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	RW	RESET			Soft reset request and status
			NoReset	0	Write to release reset
					Reading '0' means reset is not active
			Reset	1	Write to hold reset
					Reading '1' means reset is active

9.2.5.2 ERASEALL

Address offset: 0x004

Perform a secure erase of the device. The device will be returned to factory default settings upon next reset.

Bit number					31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ID					A
Reset 0x00000000					**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**
ID	Acce	Field	Value ID	Value	Description
A	W	ERASEALL			Erase flash, SRAM and UICR in sequence
			NoOperation	0	No operation
			Erase	1	Erase flash, SRAM and UICR in sequence

9.2.5.3 ERASEALLSTATUS

Address offset: 0x008

Status register for the ERASEALL operation